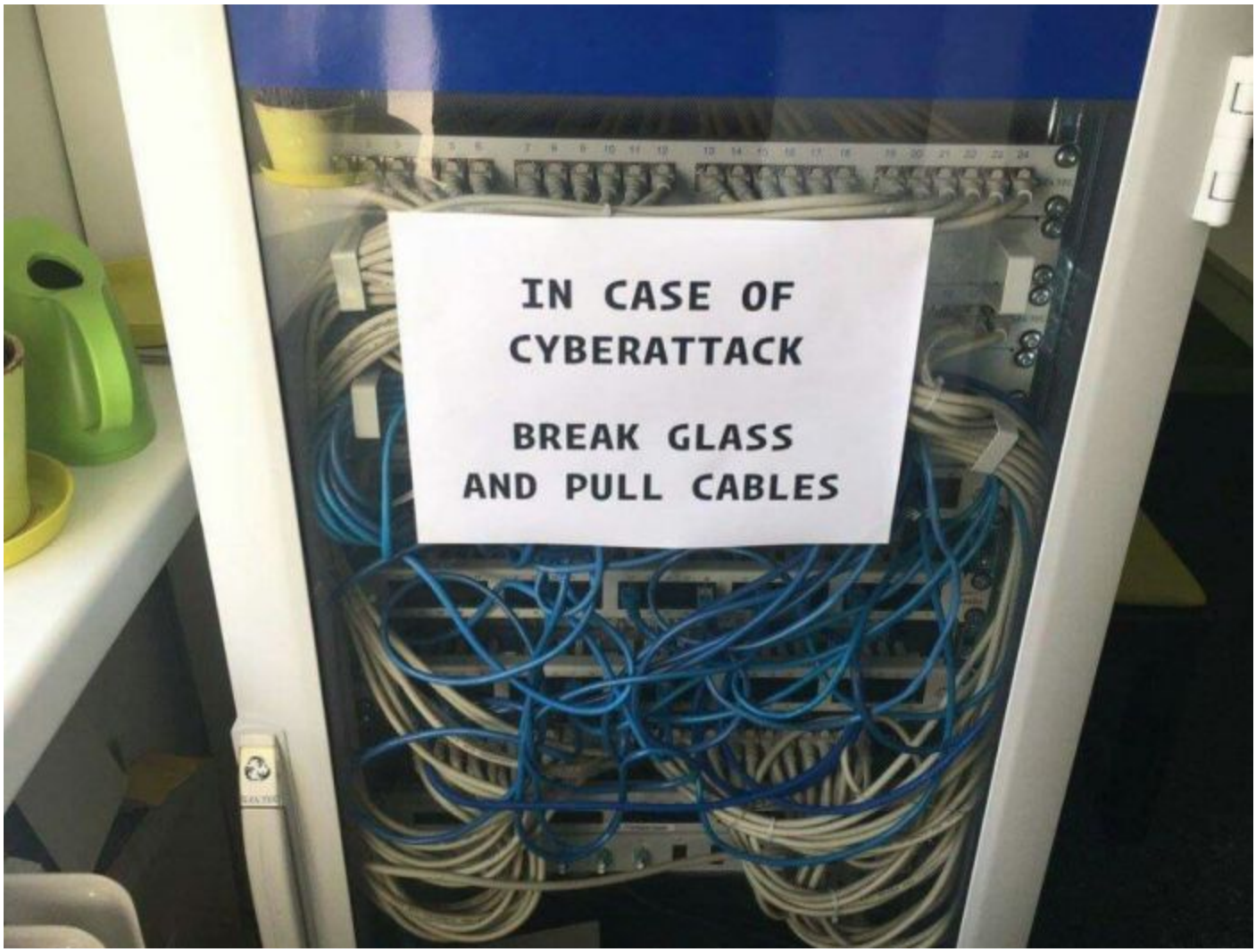
Security Now! #597 - 01-31-17

Traitors in our Midst

This week on Security Now!

The best “I'm not a Robot” video ever, Cisco's WebEx problem is far more pervasive than first believed, More bad news (and maybe some good news) for Netgear, Gmail adds .js to the no-no list, a hotel finally decides to abandon electronic room keying, more arguments against the use of modern AV, another clever exploitable CSS browser hack, some (hopefully final) password complexity follow-ups, a bit of errata and miscellany, a SQRL status update, a "Luke... trust the SpinRite" story, and a very nice analysis of a little-suspected threat hiding among us.



Security News

Robot outwits "I am not a Robot" Captcha

- https://boingboing.net/2017/01/26/robot-outwits-i-am-not-a-rob.html

- Wonderful video. (Lamest, most wonderful "arm" ever!)

- Complete with "Mic Drop" (and a brief F-bomb in the ending song lyrics.)

- While it's TRUE that a "robot" arm clicked the "I'm not a Robot" box, as we know, the only reason that checkbox offer was presented to the user in the first place was that this user's entire historical reputation with Google and the Internet, their surfing past and IP history, were all already known to NOT have a "web bot" on the end.

Cisco's WebEx browser extension trouble is/was worse than initially believed.

- https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20170124-webex

- From last week: Google's Tavis Ormandy discovered the "magic cookie URL" hiding in plain sight in Cisco's WebEx extenstion for Chrome which could allow remote code execution by causing the extension to download and execute attacker-provided code.

- Turns out the problem is much more pervasive that originally believed...

- Quote:
  A vulnerability in Cisco WebEx browser extensions could allow an unauthenticated, remote attacker to execute arbitrary code with the privileges of the affected browser on an affected system. This vulnerability affects the browser extensions for Cisco WebEx Meetings Server and Cisco WebEx Centers (Meeting Center, Event Center, Training Center, and Support Center) when they are running on Microsoft Windows.

  The vulnerability is due to a design defect in an application programing interface (API) response parser within the plugin. An attacker that can convince an affected user to visit an attacker-controlled web page or follow an attacker-supplied link with an affected browser could exploit the vulnerability. If successful, the attacker could execute arbitrary code with the privileges of the affected browser.

  Cisco has released software updates for Google Chrome, Firefox, and Internet Explorer that address this vulnerability. There are no workarounds that address this vulnerability.

  This vulnerability affects Cisco WebEx extensions and plugins for Windows when running on most supported browsers. The affected browsers are Google Chrome, Mozilla Firefox, and Internet Explorer for Windows.

  The following versions of the Cisco WebEx browser extensions are affected by the vulnerability described in this document:

    - Versions prior to 1.0.7 of the Cisco WebEx Extension on Google Chrome
    - Versions prior to 106 of the ActiveTouch General Plugin Container on Mozilla Firefox
    - Versions prior to 10031.6.2017.0126 of the GpcContainer Class ActiveX control file on Internet Explorer

- Workarounds
  There are no workarounds that address this vulnerability. However, administrators and users of Windows 10 systems may utilize Microsoft Edge to join and participate in WebEx sessions as Microsoft Edge is not affected by this vulnerability.

  Additionally, administrators and users can remove all WebEx software from a Windows system by using the Meeting Services Removal Tool, which is available from https://help.webex.com/docs/DOC-2672.

CVE-2017-5521: Bypassing Authentication on NETGEAR Routers

- https://www.trustwave.com/Resources/SpiderLabs-Blog/CVE-2017-5521--Bypassing-Authentication-on-NETGEAR-Routers/

- Why does this matter?
  Quote: "For starters, it affects a large number of models. We have found more than ten thousand vulnerable devices that are remotely accessible. The real number of affected devices is probably in the hundreds of thousands..."

- (Yesterday) Simon Kenin posted to TrustWave's SpiderLabs Blob:
  Home routers are the first and sometimes last line of defense for a network. Despite this fact, many manufacturers of home routers fail to properly audit their devices for security issues before releasing them to the market. As security researchers, we are often disappointed to rediscover that this is not always the case, and that sometimes these vulnerabilities simply fall into our hands during our day-to-day lives.

  Such is the story of the two NETGEAR vulnerabilities I want to share with you today:

  It was a cold and rainy winter night, almost a year ago, when my lovely NETGEAR VEGN2610 modem/router lost connection to the Internet. I was tucked in bed, cozy and warm, there was no way I was going downstairs to reset the modem, "I will just reboot it through the web panel" I thought to myself. Unfortunately I couldn't remember the password and it was too late at night to check whether my roommates had it.

  I considered my options:

    - Get out of bed, go downstairs and freeze as I reboot the router.
    - Be lazy, stay in bed, and since I am a security researcher, try to hack it :)

  Needless to say, I chose the latter. "So… where do I start?" I thought to myself, "Well, it has a web interface and I need to bypass the authentication somehow, so the web server is a good start."

I started manually fuzzing the web server with different parameters, I tried "../.." classic directory traversal and such, and after about 1 minute of fuzzing, I tried "…" and I got this response:

```
<html>
<head>
<meta http-equiv='Content-Type' content='text/html; charset=utf-8'>
<title>401 Unauthorized</title></head>
<body onload="document.aForm.submit()"><h1>401 Unauthorized</h1>
<p>Access to this resource is denied, your client has not supplied the correct authentication.</p>
<form method="post" action="unauth.cgi?id=1211868232" name="aForm"></form></body>
</html>
```

"Hmm, what is that unauth.cgi thingy? and what does that id number mean?", I thought to myself.

Luckily for me the Internet connection had come back on its own, but I was now a man on a mission, so I started to look around to see if there were any known vulnerabilities for my VEGN2610. It turned out that there are none. :<

I started looking up what that "unauth.cgi" page could be, and I found 2 publicly disclosed exploits from 2014, for different models that manage to do unauthenticated password disclosure. Booyah! Exactly what I need. (link 1 & link 2)

Those two guys found out that the number we get from unauth.cgi can be used with passwordrecovered.cgi to retrieve the credentials.

I tested the method described in both, and voila - I have my password, now I can go to sleep happy and satisfied.

I woke up the next morning excited by the discovery, I thought to myself: "3 routers with same issue… Coincidence? I think not". Luckily, I had another, older NETGEAR router laying around; I tested it and bam! Exploited.

I started asking people I knew if they have NETGEAR equipment so I could test further to see the scope of the issue. In order to make life easier for non-technical people I wrote a python script to test for this issue.

I am not a great programmer. I am aware of that, and that is why I don't work as a full time programmer. As it turned out, I had an error in my code where it didn't correctly take the number from unauth.cgi and passed gibberish to passwordrecovered.cgi instead, but somehow it still managed to get the credentials!

"Wait… what is going on here?" I thought to myself. After few trials and errors trying to reproduce the issue, I found that the very first call to passwordrecovered.cgi will give out the credentials no matter what the parameter you send. This is totally new bug that I haven't seen anywhere else. When I tested both bugs on different NETGEAR models, I found that my second bug works on a much wider range of models.

A full description of both of these findings as well as the python script used for testing is available online and the vulnerabilities have been assigned CVE designations.

The Responsible Disclosure Process

This is where the story of discovery ends and the story of disclosure begins. Following our Responsible Disclosure policy we sent both findings to NETGEAR in the beginning of April 2016.

In our initial contact, the first advisory had 18 models listed as vulnerable, although six of them didn't have the vulnerability in the latest firmware. Perhaps it was fixed as part of a different patch cycle. The second advisory included 25 models, all of which were vulnerable in their latest firmware version.

In June NETGEAR published a notice that provided a fix for a small subset of vulnerable routers and a workaround for the rest. They also made the commitment to working toward 100% coverage for all affected routers. The notice has been updated several time since then and currently contains 31 vulnerable models, 18 of which are patched now, and 2 models that they previously listed as vulnerable, but are now listed as not vulnerable. In fact, our tests show that one of the models listed as not vulnerable (DGN2200v4) is, in fact, vulnerable and this can easily be reproduced with the POC provided in our advisory.

Over the past nine months we attempted to contact NETGEAR multiple times for clarification and to allow them time to patch more models. Over that time we have found more vulnerable models that were not listed in the initial notice, although they were added later. We also discovered that the Lenovo R3220 router is powered by NETGEAR firmware and it was vulnerable as well.

Luckily NETGEAR did eventually get back to us right before we were set to disclose these vulnerabilities publicly. We were a little skeptical since our experience to date matched that of other third-party vulnerability researchers that have tried to responsibly disclose to NETGEAR only to be met with frustration.

Two changes helped sway our opinion.

The first was that NETGEAR committed to pushing out firmware to the currently unpatched models on an aggressive timeline.

The second change made us more confident that NETGEAR was not just serious about patching these vulnerabilities, but serious about changing how they handle third-party disclosure in general. That change was their commitment to Bugcrowd (https://bugcrowd.com/netgear), a popular third-party vendor that helps to vet research, provides oversight for the patching process and provides bug bounty rewards to help to motivate third-party researchers. We fully expect this move will not only smooth the relationship between third-party researchers and NETGEAR, but, in the end, will result in a more secure line of products and services.

Why Is This Vulnerability So Critical?

For starters, it affects a large number of models. We have found more than ten thousand vulnerable devices that are remotely accessible. The real number of affected devices is probably in the hundreds of thousands, if not over a million.

The vulnerability can be used by a remote attacker if remote administration is set to be Internet facing. By default this is not turned on. However, anyone with physical access to a network with a vulnerable router can exploit it locally. This would include public wifi spaces like cafés and libraries using vulnerable equipment.

As many people reuse their password, having the admin password of the router gives us an initial foothold on the network. We can see all the devices connected to the network and try to access them with that same admin password.

With malware such as the Mirai botnet being out there, it is also possible that some of the vulnerable routers could be infected and ultimately used as bots as well. If running a bot is not possible, the DNS can be easily changed to a rogue one, as described by Proofpoint, to further infect machines on the network.

We recommend that all users of NETGEAR equipment check the Knowledge Base Article for instructions to test if you are vulnerable and how to apply patched firmware if you are.

Gmail will block .js file attachments starting February 13, 2017

- http://gsuiteupdates.googleblog.com/2017/01/gmail-will-restrict-js-file-attachments.html

- Gmail currently restricts certain file attachments (e.g. .exe, .msc, and .bat) for security reasons, and starting on February 13, 2017, we will not allow .js file attachments as well. Similar to other restricted file attachments, you will not be able to attach a .js file and an in-product warning will appear, explaining the reason why. For inbound mail, senders will get a bounce message explaining why the email was blocked. If you still need to send .js files for legitimate reasons, you can use Google Drive, Google Cloud Storage, or other storage solutions to share or send your files.

- To prevent against potential viruses, Gmail doesn't allow you to attach certain types of files, including:

    - .ADE, .ADP, .BAT, .CHM, .CMD, .COM, .CPL, .EXE, .HTA, .INS, .ISP, .JAR, .JSE, .LIB, .LNK, .MDE, .MSC, .MSI, .MSP, .MST, .NSH, .PIF, .SCR, .SCT, .SHB, .SYS, .VB, .VBE, .VBS, .VXD, .WSC, .WSF, .WSH

    - And their compressed form (like .gz or .bz2 files) or when found within archives (like .zip or .tgz files)
    - Documents with malicious macros
    - Archives whose listed file content is password protected
    - Archives whose content includes a password protected archive

Hotel ransomed by hackers as guests locked out of rooms

- http://www.thelocal.at/20170128/hotel-ransomed-by-hackers-as-guests-locked-in-rooms

- Widely misreporting claimed that guests were also locked IN their rooms.

- 111 year old, high-end luxury 4-Star Austrian hotel, attacked by cryptomalware three times. Each time they had no choice but to pay the ransom.

- Managing Director Christoph Brandstaetter said: "The house was totally booked with 180 guests, we had no other choice. Neither police nor insurance help you in this case."

  "The restoration of our system after the first attack in summer has cost us several thousand Euros. We did not get any money from the insurance so far because none of those to blame could be found."

- The manager said it was cheaper and faster for the hotel to just pay the Bitcoin.

- Brandstaetter said: "Every euro that is paid to blackmailers hurts us. We know that other colleagues have been attacked, who have done similarly."

- When the hackers got the money, they unlocked the key registry system and all other computers, making them all run as normal again.

- Yet according to the hotel, the hackers left a back door open in the system, and tried to attack the systems again.

- Brandstaetter said: "We are planning at the next room refurbishment for old-fashioned door locks with real keys. Just like 111 years ago at the time of our great-grandfathers."

"It might be time to stop using antivirus"

- Update your software and OS regularly instead, practice skeptical computing.
- https://arstechnica.com/information-technology/2017/01/antivirus-is-bad/

- ArsTechnica, last Friday, by Sebastian Anthony in the UK
  Former Firefox developer Robert O'Callahan, now a free agent and safe from the PR tentacles of his corporate overlord, says that antivirus software is terrible, AV vendors are terrible, and that you should uninstall your antivirus software immediately—unless you use Microsoft's Windows Defender, which is apparently okay.

  A couple of months back, Justin Schuh, Google Chrome's security chief, and indeed one of the world's top infosec bods, said that antivirus software is "my single biggest impediment to shipping a secure browser." Further down the thread he explains that meddling AV software delayed Win32 Flash sandboxing "for over a year" and that further sandboxing efforts are still on hold due to AV. The man-in-the-middle nature of antivirus also causes a stream of TLS (transport layer security) errors, says Schuh, which in turn breaks some elements of HTTPS/HSTS.

These are just two recent instances of browser makers being increasingly upset with antivirus software. Back in 2012, Nicholas Nethercote, another Mozillian working on Firefox's MemShrink project said that "McAfee is killing us." In that case, Nethercote was trying to reduce the memory footprint of Firefox, and found that gnarly browser add-ons like McAfee were consuming a huge amount of memory, amongst other things. If you venture off-piste into the browser mailing lists, anti-antivirus sentiment has bubbled away just below the surface for a very long time.

The problem, from the perspective of the browser makers, is that antivirus software is incredibly invasive. Antivirus, in an attempt to catch viruses before they can infect your system, forcibly hooks itself into other pieces of software on your computer, such as your browser, word processor, or even the OS kernel. O'Callahan gives one particularly egregious example: "Back when we first made sure ASLR was working for Firefox on Windows, many AV vendors broke it by injecting their own ASLR-disabled DLLs into our processes." ASLR, or address-space layout randomisation, is one of the better protections against buffer overflow exploits.

- And we have discussed here that modern A/V is introducing a larger and more fertile attack surface than the malware they're attempting to curtail.

- A/V integrated into the OS by the vender has become the only right answer.

Look before you paste from a website to terminal

- http://lifepluslinux.blogspot.co.nz/2017/01/look-before-you-paste-from-website-to.html

- Looks like an innocent command: ls -lat

- Actually contains:
  ls ; clear; echo 'Haha! You gave me access to your computer with sudo!'; echo -ne 'h4cking ## (10%)\r'; sleep 0.3; echo -ne 'h4cking ### (20%)\r'; sleep 0.3; echo -ne 'h4cking ##### (33%)\r'; sleep 0.3; echo -ne 'h4cking ####### (40%)\r'; sleep 0.3; echo -ne 'h4cking ######### (50%)\r'; sleep 0.3; echo -ne 'h4cking ############ (66%)\r'; sleep 0.3; echo -ne 'h4cking ###################### (99%)\r'; sleep 0.3; echo -ne 'h4cking ######################## (100%)\r'; echo -ne '\n'; echo 'Hacking complete.'; echo 'Use GUI interface using visual basic to track my IP'
  ls -lat

- Very much like the form-obfuscation hack... uses CSS to make content invisible... yet still operable.

Follow-Ups

- @SGgrc just listened to SN.
  How strong do pronounsable PWs calculate, Lastpass feature?
  I use for few I must remember.

- Max : @SGgrc You go on about logarithms for half an hour, but you fail to actually answer the question. God that is so frustrating to listen to!

Errata

- We have listeners among Google's security team who reached out to add a bit of additional information about the troubling auto-form-fill hack: The CC number field will NOT be populated until and unless the user responds with the card's correct CVV (card verification value).

- Taylor Hornby (defuse.ca) @SGgrc Quantum entanglement doesn't let you talk FTL but you can use it to get one-time-pad keys.

Miscellany

- Addam Tait: @SGgrc Curious about your opinion re:software delivery teams.
  Better to have security person per team overseeing, or all devs taking ownership of security

- Christian Loris: @SGgrc Heard your ponder on password monkey's prevalence - maybe part French? Mon Key = My Key!

SQRL Update

- Manage Shared Access
- Tightly-bound identity is where the world is headed.
- "Sharing" logons will be less easy and more actively thwarted in the future.
- SQRL is inherently a tightly-bound identity system.
- Deliberately shared SQRL identities ARE possible, but are not the proper solution.
- So from the start GRC will be providing a working model of a many-to-one identity mapping solution which we call "Managed Shared Access."
- This makes account sharing and account transfer simple, clean, transparent, and auditable.

SpinRite

From: Pete Kokkinis
Subject: Please forward to Steve - testimonial

Hey Steve I'm a long time SN listener and user of SpinRite. You are an inspiration to me and I look forward to your weekly podcasts - BUT for approximately 14 minutes, I doubted SpinRite and for that I'd like to apologize. I had a PC running at a client site acting as a syslog'er for many years. Unbeknownst to me, my client closed up shop and moved when I went to visit their shared space with another tenant client of mine.

So I picked up some things I left behind, including the syslog PC. I fired it up back at my office only to find it hanging at bootup with a message about HAL.DLL missing or corrupt. Being curious at this point, I ran SpinRite. It's forward progress stalled at 4%. Since I was impatient I canceled it and tried booting as ANY running of SpinRite often seems to get enough of the disk back. This time it didn't. So I reran SpinRite, starting it at 5% and it completed in an hour. I tried booting again, nothing.

So I plugged the drive into a USB/SATA adapter to see if my Syslog folder could simply be copied off. No dice. Even though I could see the folder, I couldn't even open it as it would just hang. So I right clicked the drive and ran a CHKDSK with both options checked.

[x] Automatically fix file system errors
[x] Scan for and attempt recovery of bad sectors

The drive was making a normal scanning noise as CHKDSK was running, and the progress bar was steadily moving. At this point I questioned the usefulness of SpinRite if CHKDSK can do this recovery more successfully. The progress bar finally reached the end after 14 short minutes and I clicked Close.

I browsed to the external drive letter, opened Program Files, and BAM! Half the folders in the system were completely gone, including the Syslog folder I needed and had I seen earlier!

I Googled if CHKDSK can delete files and wow did I see some upset users out there!

It never occurred to me that if chkdsk can't repair, it will wipe data if corrupt granted you check both option boxes.

Anyway, keep up the great work and I will never doubt you again!

Are ubiquitous Printers traitors in our midst?

http://web-in-security.blogspot.de/2017/01/printer-security.html

Security Research:
https://www.nds.rub.de/research/publications/sok-exploiting-network-printers/

Jens Müller's Masters Thesis: "Exploiting Network Printers - A Survey of Security Flaws in Laser Printers and Multi-Function Devices"
https://www.nds.rub.de/media/ei/arbeiten/2017/01/30/exploiting-printers.pdf

- Introduction:
  The paperless office has been a dream for more than three decades. However, nowadays printers are still one of the most essential devices for daily work and common Internet users. Instead of getting rid of them, printers evolved from simple printing devices to complex network computer systems installed directly in company networks, and carrying lots of confidential data in their print jobs. This makes them to an attractive attack target.

Attack Categories		Print Job Manipulation		Information Disclosure						# Printer Vulnerabilities
Attacks		content overlay	content replacement	memory access	file system access		print job capture	credential disclosure		
Printers / Printer Languages		PS		PJL	PS	PJL	PS	PS	PJL	
1	HP LaserJet 1200	1	1				1	1*	1	7
2	HP LaserJet 4200N	1	1		1	1	1	1*	1	12
3	HP LaserJet 4250N	1	1		1	1	1	1*	1	12
4	HP LaserJet P2015dn	1	1				1	1*	1	10
5	HP LaserJet M2727nfs	1	1				1	1*	1	10
6	HP LaserJet 3392 AiO	1	1				1	1*	1	10
7	HP Color LaserJet CP1515n	1	1				1	1*	1	10
8	Brother MFC-9120CN			1	1*			1	1	7
9	Brother DCP-9045CDN			1	1*			1	1	7
10	Lexmark X264dn	1	1		1*		1	1*	n/a	9
11	Lexmark E360dn	1	1		1*		1	1*	n/a	10
12	Lexmark C736dn	1	1		1*		1	1*	n/a	10
13	Dell 5130cdn	?	?		1*		1	1*	n/a	5
14	Dell 1720n	1	1		1*		1	1*	n/a	11
15	Dell 3110cn	1	1			1*			n/a	6
16	Kyocera FS-C5200DN	1	1		1*			n/a	1	8
17	Samsung CLX-3305W	?	?						n/a	1
18	Samsung MultiPress 6345N	?	?						n/a	1
19	Konica bizhub 20p			1	1*			1	1	7
20	OKI MC342dn	1	1		1*	1*	1	1*	n/a	8
	# Vulnerable Printers	14	14	3	12	4	13	16	11	

Legend:
- 1 — device vulneral
- 1* — vulnerability is
- (empty) — not vulnerable/PostScript feedback not available

Figure 7: Results of our evaluation show that a majority of common printer devices is vulnerable to the analyzed attacks.

- And…
  In this paper we conduct a large scale analysis of printer attacks and systematize our knowledge by providing a general methodology for security analyses of printers. Based on our methodology we implemented an open-source tool called PRinter Exploitation Toolkit (PRET). We used PRET to evaluate 20 printer models from different vendors and found all of them to be vulnerable to at least one of the tested attacks. These attacks included, for example, simple Denial-of-Service (DoS) attacks or skilled attacks extracting print jobs and system files.

  On top of our systematic analysis we reveal novel insights that enable attacks from the Internet by using advanced cross-site printing techniques combined with printer CORS-Spoofing. Finally, we show how to apply our attacks to systems beyond typical printers like Google Cloud Print or document processing websites. We hope that novel aspects from our work will become the foundation for future researches, for example, for the analysis of IoT security.

- Although the potential threat has been known for more than 20 years, printers are unglamorous utilitarian workhorses lacking large screens and keyboards, so they're easily ignored. (Remember that when Apple introduced their famous laser printer the CPU inside was far more powerful than the computers driving it.)

- Even though many proof-of-concept attacks and techniques are known for years, the according countermeasures have not been implemented, leaving the devices and systems vulnerable.

- What's the #1 problem??
  - Printers are crammed with a large collection of legacy INTERPRETERS.

- Protocols:
  - IPP - Internet Printing Protocol
  - LPD - Line Printer Daemon
  - SMB - Server Message Block
  - SNMP - Simple Network Management Protocol
  - Raw Port 9100 printing (JetDirect and AppSocket)

- Even though many proof-of-concept attacks and techniques are known for years, the according countermeasures have not been implemented, leaving the devices and systems vulnerable.

- Job / Printer Control Languages:
  - Printer Job Language (PJL). PJL was originally introduced by HP but soon became a de-facto standard for print job control. PJL "resides above other printer languages" and can be used to change settings like paper tray or size.

  - Printer Management Language (PML). PML is a proprietary language to control HP printers. It basically combines the features of PJL and SNMP.

- Page Description Languages:
    - There are various proprietary page description languages like…
    - Kyocera's PRESCRIBE
    - Samsung Printer Language (SPL)
    - Xerox Escape Sequence (XES)
    - Canon Printing System Language (CaPSL)
    - Ricoh Refined Printing Command Stream (RPCS)
    - Epson Standard Code for Printers (ESC/P)
    - Hewlett-Packard Graphics Language (HP-GL) and HP-GL/2
        - (which have been designed for plotters)

    - Support for direct Portable Document Format (PDF) and
    - XML Paper Specification (XPS) printing is also common on newer printers.

    - The most common 'standard' page description languages however are the Printer Command Language (PCL) (which is hard to exploit from a security perspective due to its limited capabilities) and PostScript.

- Postscript:
    - PostScript is a stack-based, turing-complete programming language consisting of about 400 operators for arithmetics, stack and graphic manipulation and various data types such as arrays or dictionaries.

    - Technically, access to a PostScript interpreter can already be classified as code execution because any algorithmic function can theoretically be implemented in PostScript.

- Three Attack Models:
    - **A local attacker** is the strongest attacker, having physical access to the printer for a limited time. The attacker's capabilities include:
        - Plugging in external storage media like memory cards or USB sticks
        - Temporarily connecting to the printer device via USB or parallel cable
        - Changing control panel settings and pressing certain key combinations.

    - And this is a realistic attack for most institutions and companies. Gaining physical access to printer devices can generally be considered easier it is for other network components like servers or workstations because printers are usually explicitly shared by and accessible to a whole department. Sneaking into an unlocked copy room and launching a malicious print job from USB stick takes only a matter of seconds.
    - **A LAN attacker** can connect to a printer device via a TCP/IP network is capable of accessing all network services offered by the device, including but not limited to web, FTP, SMB, SNMP, LPD, IPP, or raw port 9100 printing and establishing various connections over a longer period.

    - Many newer printers bring their own wireless access point to allow easy printing, for example, via AirPrint compatible mobile apps. While connecting to a printer through Wi-Fi requires the attacker to stay physically close to the device, it may be feasible

to perform an attack from outside of the targeted institution depending on the signal strength.

- **A Web Attacker** -- It's actually possible to perform "Cross Site Printing" attacks!

- The so called cross-site printing technique is directly related to this attacker model and enables the execution of different attacks even outside the network where the printer is located. Cross-site printing is used as a carrier for the attack vectors.

- The only requirement in this attacker model is that a web attacker controls the content of a website and is able to lure a victim to this website. By visiting the website, the attacker can deploy JavaScript code to be processed by the victim's web browser. Thus, the attacker initiates AJAX requests to port 9100 of the victim's intranet printer and sends raw PostScript or PJL commands. Consequently, the printer executes the malicious code. This way the attacker can reach even printers which are not directly visible from the Internet.

- Responsible disclosure:
  These researchers responsibly disclosed all security vulnerabilities to printer manufacturers and to administrators responsible for vulnerable interpreter processing websites.

  Google rewarded their findings with $3133.7.