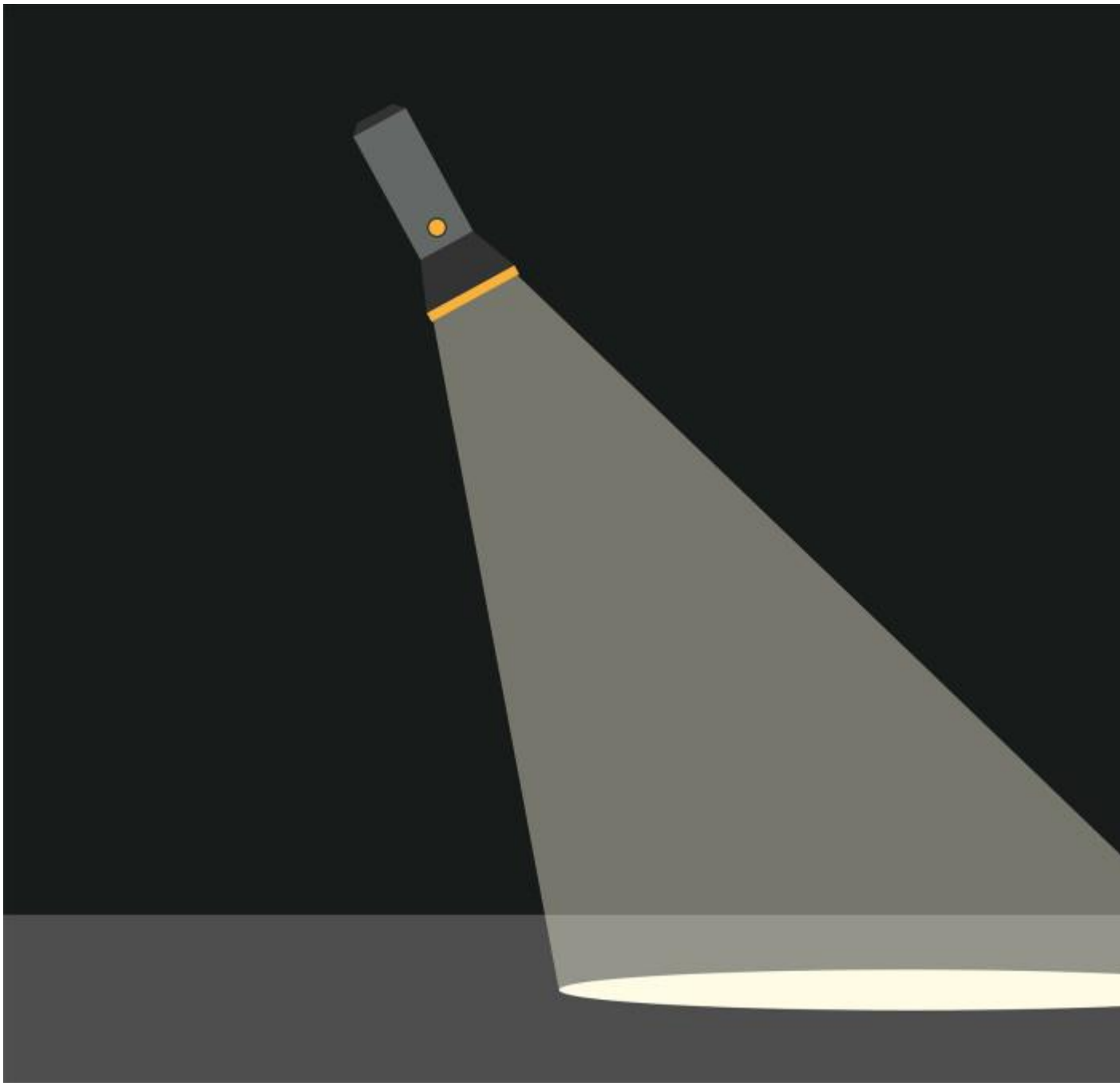
- Author: Kim Zetter. Kim Zetter Security
- Date of Publication: 01.13.16. 01.13.16
- Time of Publication: 9:00 am. 9:00 am

Hacking Team’s Leak Helped Researchers Hunt Down a Zero-Day



Click to Open Overlay Gallery Getty Images

Zero-day exploits are a hacker's best friend. They attack vulnerabilities in software that are unknown to the software maker and are therefore unpatched. Criminal hackers and intelligence agencies use zero day exploits to open a stealth door into your system, and because antivirus companies also don't know about them, the exploits can remain undetected for years before they're discovered. Until now, they've usually been uncovered only by chance.

But researchers at Kaspersky Lab have, for the first time, discovered a valuable zero-day exploit after intentionally going on the hunt for it. And they did so by using only the faintest of clues to find it.

The malware they found is a remote-code execution exploit that attacks a vulnerability in Microsoft's widely used Silverlight software—a browser plug-in Netflix and other providers use to deliver streaming content to users. It's also used in SCADA and other industrial control systems that are installed in critical infrastructure and industrial facilities.

The vulnerability, which Microsoft called "critical" in a patch released to customers on Tuesday, would allow an attacker to infect your system after getting you to visit a malicious website where the exploit resides—usually through a phishing email that tricks you into clicking on a malicious link. The attack works with all of the top browsers except Chrome—but only because Google removed support for the Silverlight plug-in in its Chrome browser in 2014.

Kaspersky Lab caught its big fish, the Silverlight exploit, in late November after the zero-day infected a customer's machine. But it took a clever lure and months of patient waiting to get that prize. The story behind that discovery provides an intriguing lesson in how researchers might uncover more zero days hidden in the wild.

Hacking Team's Hacked Emails Offered the First Clue

It all began with a conversation that was never meant to be public.

In July 2015, a hacker known only as "Phineas Fisher" targeted the Italian surveillance firm Hacking Team and stole some 400 GB of the company's data, including internal emails, which he dumped online. The hack exposed the company's business practices, but it also revealed the business of zero-day sellers who were trying to market their exploits to Hacking Team. The controversial surveillance firm, which sells its software to law enforcement and intelligence agencies around the world—including to oppressive regimes like Sudan, Bahrain, and Saudi Arabia—uses zero-day exploits to help sneak its surveillance tools onto targeted systems.

Costin Raiu, head of Kaspersky's Global Research and Analysis Team, became intrigued by one negotiation in particular that occurred in 2013 between Hacking Team and a zero-day seller who identified himself as a 33-year-old Russian named Vitaliy Toropov. In a series of emails dumped online and highlighted in an *Ars Technica* story about the hacked data, the researcher negotiated the successful sale of a $45,000 Flash exploit to Hacking Team.

After completing negotiations on that exploit, Toropov, like all good businessmen, tried to interest Hacking Team in more of his goods, which he was willing to sell at a discount for bulk buys—a $5,000 discount if Hacking Team purchased a second zero day from him, and a $10,000 discount if they purchased a third. Among his offerings: "I recommend you the fresh 0day for iOS 7/OS X Safari," he wrote, "or my old Silverlight exploit which was written 2.5 years ago and has all chances to survive further in next years as well."

Although the iOS exploit was interesting, Raiu was much more intrigued by the Silverlight exploit that Toropov said had remained undetected since 2011. It wasn't an idle boast from an inexperienced newcomer.

Toropov is a prolific bug hunter and exploit writer who until 2013 was an active participant in bug bounty programs—programs that pay bug hunters money for information about vulnerabilities they find, which is then passed to the software makers so they can patch the holes. Between 2011 and 2013, Toropov disclosed more than 40 vulnerabilities to these programs, according to a spreadsheet he has published online and a page for his discoveries on the Packet Storm security site. But in October 2013, his public disclosure of bugs dried up after he disclosed two vulnerabilities in Silverlight to Microsoft. That same month is when he began secretly marketing his wares to Hacking Team—including, apparently, one Silverlight exploit he'd kept from Microsoft in order to sell it to customers who would use it to hack systems.

If the exploit had already been sold to other customers and was infecting systems in the wild for two and a half years, Raiu wondered if he might be able to find it. There was just one problem. Toropov provided no details about the exploit that might help him track it down.

Usually zero days are found by accident when someone discovers they've been hacked, and a forensic examination of their system uncovers zero-day malware. Once these exploits are discovered, antivirus companies look for tell-tale fingerprints in the code that can help them locate the malware on other systems; then they write signatures their scanners use to search customer systems. But in this case, Raiu wasn't sure how to look for the zero-day exploit since he didn't have code to examine and didn't even know what vulnerability in Silverlight it targeted.

Click to Open Overlay GalleryCostin Raiu, head of Kaspersky Lab's Global Research and Analysis Team. Kaspersky Lab

But after looking at Toropov's public list of previous bug discoveries, he got an idea. He started examining the proof-of-concept exploit Toropov had written for the bugs he'd already discovered to see if he might find any particular programming techniques or patterns in the

way he wrote code that could be used as a signature to find exploits of his that might be in the wild. Researchers provide proof-of-concept exploits to bug bounty programs to verify in a benign way that the vulnerabilities they've found are real and can be exploited. Usually the proof-of-concept code simply launches the calculator application on a machine to provide visual proof that the exploit worked.

Raiu's instinct about looking at the published files was right. He examined in particular some proof-of-concept code Toropov had published for one of the Silverlight vulnerabilities Microsoft had patched in 2013. Among the files for this exploit was one that contained debugging code. Debugging code is used by developers to look for errors in their program as they're writing it. There were three particular strings of debugging code that caught Raiu's eye that appeared in multiple files Toropov wrote.

"With exploit developers they have [code] libraries they build and they keep reusing them from one exploit to another in order to simplify their work," Raiu notes. "I said, what if his other Silverlight exploits are similar to this proof-of concept one he wrote in 2013?"

More on Zero-Day Exploits

- 

  An Unprecedented Look at Stuxnet, the World's First Digital Weapon

- 

  Hacker Lexicon: What Is a Zero Day?

- **Hackers Claim Million-Dollar Bounty for iOS Zero Day Attack**

Programmers usually take debugging code out of the final versions of their programs, but sometimes they leave it in the source code and it gets compiled into the binary, even though it's not code that gets used by the exploit to perform its functions. Raiu was hoping that was the case.

He used a tool called YARA to see if he could find traces of the strings on Kaspersky customer systems. YARA was designed in 2007 by Victor Manuel Alvarez, a Spanish security researcher who works at VirusTotal, a free online virus scanner that Google now owns. Using the tool, researchers can create a so-called YARA rule to search for malicious files and uncover patterns in them in order to group similar files into families of malware. YARA rules can also be used to scan networks and systems for the same patterns of code. That's how Raiu decided to use it.

He'd tried to use YARA rules once before in this way, but had failed to find what he was seeking. One of Kaspersky's customers had been attacked by two exploits, which came in through an infected Adobe .PDF file. One of the exploits allowed the attackers to escape from the Adobe Reader sandbox—a protective layer some vendors put in their software to prevent exploits from jumping out of an application and infecting the core system. Raiu and his colleagues never found the exploits, but were able to figure out how they worked and notified vendors to get the vulnerable holes patched.

Despite that previous failure, Raiu thought it was worth trying a YARA rule again with Toropov's exploit. In July, shortly after reading the emails Toropov exchanged with Hacking Team, Raiu created a YARA rule based on the debugging code he'd found and then distributed it to the company's automatic exploit prevention tool and the Kaspersky Security Network, composed of customers who have opted to share with Kaspersky malicious samples found on their systems. Then he waited.

```
rule exploit_Silverlight_Toropov_Generic_XAP {

meta:

    author = "Kaspersky Lab"
    filetype = "Win32 EXE"
    date = "2015-07-23"
    version = "1.0"

strings:


    $b2="Can't find Payload() address"  ascii wide
    $b3="/SilverApp1;component/App.xaml" ascii wide
    $b4="Can't allocate ums after buf[]" ascii wide
    $b5="------------ START ------------"

condition:

    ( (2 of ($b*)) )

}
```

Click to Open Overlay GalleryDebugging strings in the YARA rule Kaspersky used to find the Silverlight exploit. Image courtesy of Kaspersky Lab

Months passed and there was no sign of an infection for any customers. Raiu eventually forgot about his little experiment.

Then on November 25th an infection suddenly popped up on a customer's machine in the Middle East. Customers in the company's KSN network agree to allow malicious code found on their machines to be sent to Kaspersky for analysis. Notably, a few hours later, someone uploaded a sample of the same exploit to the Virus Total web site, but from a different geographic region. Virus total is a site that aggregates multiple virus scanners so people can upload suspicious files to it and determine if they're malicious. The file was uploaded from an IP address in Laos. It had been compiled on July 21, just a couple of weeks after Toropov's emails with Hacking Team discussing his Silverlight exploit had been exposed online.

It didn't take long, once Raiu and his team got their hands on their customer's malicious code, to determine that it was indeed a Silverlight zero day exploit.

"These particular debug strings were the only thing we could hang onto from his [earlier] Silverlight exploits," he says. Odds were against his gamble working; but it did.

Since then, Kaspersky hasn't uncovered any other samples on customer machines, which suggests whoever was using the exploit was using it judiciously to target only specific victims. The fact that two victims in different parts of the world were apparently hit on the same day suggests the attacker was conducting a campaign on that day targeting various

victims at the same time. Raiu estimates the exploit was worth between $20,000 and $40,000 on the zero-day market.

WIRED reached out to Toropov about the exploit to ask if he had written it, and passed him the technical description that Kaspersky had written about the vulnerability it targets—a BinaryReader bug in the Silverlight software. He said he wasn't familiar with the vulnerability.

"I didn't [know] about this particular BinaryReader bug," he wrote in a message to WIRED. He asked if the exploit included code from any of his previous exploits, and when told that it did, he asked to see it. WIRED sent him the code after Microsoft had already distributed its patch for the vulnerability.

"I would like to have this 0day, but unfortunately it's not mine," he said after examining it. "Anyway it was interesting to find the parts of my calc poc in this shellcode, thanks for sharing."

His term "calc poc" refers to the calculator proof-of-concept code he had published in 2013 for the previous Silverlight vulnerability Microsoft had patched back then.

Toropov didn't say why proof-of-concept code he wrote was showing up in an exploit he says he didn't write, but he said he wasn't surprised to see it in the exploit Kaspersky found. Asked if he ever ended up selling Hacking Team the Silverlight exploit he offered them in his 2013 email, he said no.

Raiu says it doesn't make sense that someone else would have put Toropov's public proof-of-concept code in their exploit, but it's not out of the question. He saw it happen in at least one other case when someone used parts of the proof-of-concept code Toropov wrote for the 2013 Silverlight vulnerability he had disclosed to Microsoft, and used that as a building block to create an exploit.

Whether or not the exploit was written by Toropov, Raiu considers his hunt for it to be a big success, since there's one less zero-day vulnerability available for attackers to exploit.

"This is actually the first time that we are succeeding in catching something that we planned on hunting," Raiu says. "It was probably a bit of intuition and luck. If the compiler would have removed these [debugging] strings, then obviously [there would have been] no luck for me."

But now that the technique has proven successful, it may be possible to examine code from other Toropov exploits to uncover additional zero days that may be using it. And if there are similar tell-tale signs in the public code of other researchers, this may be used to uncover more zero-day exploits as well.

http://www.wired.com/2016/01/hacking-team-leak-helps-kaspersky-researchers-find-zero-day-exploit/