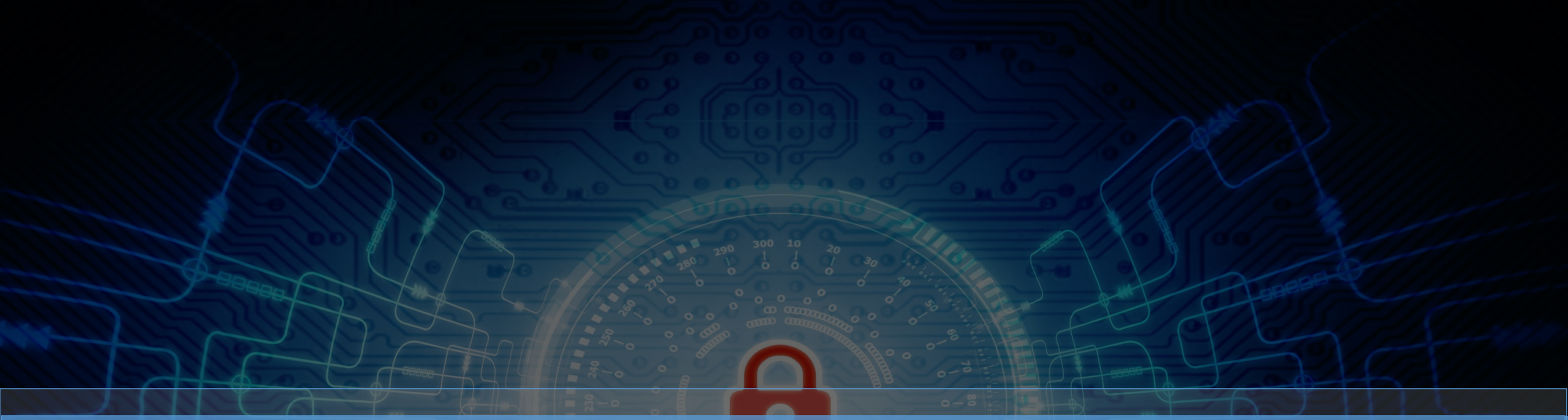
CHAPTER 5

Web and Database Attacks



Episode 5.01 – OWASP Top 10

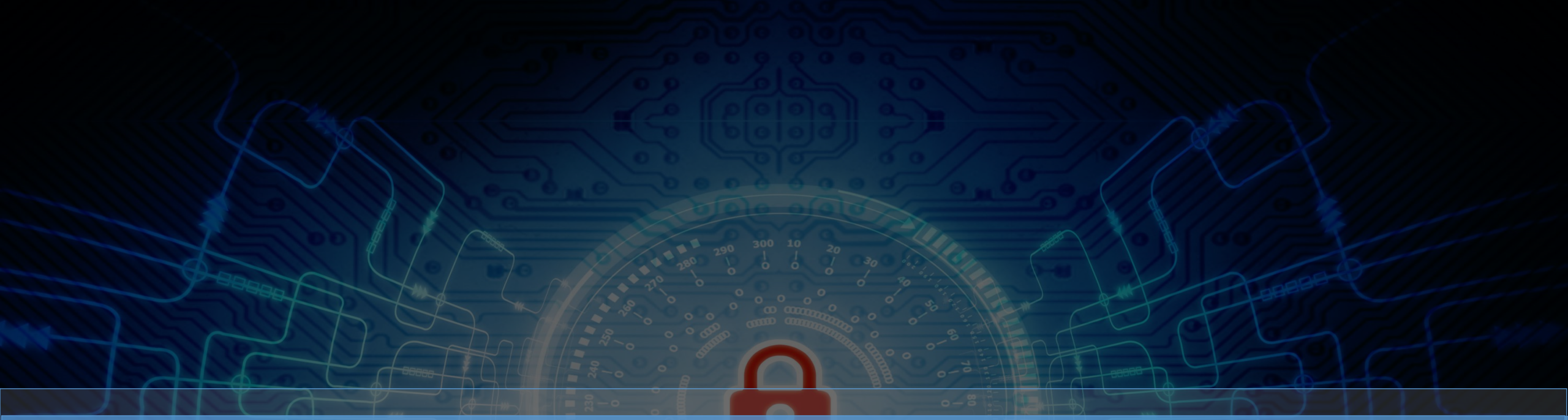
Objective 3.3 Given a scenario, research attack vectors and perform application-based attacks

OWASP TOP TEN

- Open Web Application Security Project (OWASP)
- Top ten most commonly encountered web application risks
- https://owasp.org/Top10/
- Interesting to see how many risks persist year over year

QUICK REVIEW

- OWASP identifies weaknesses in most applications
- OWASP Top Ten is a great starting point for securing software



Episode 5.02 – Application Exploits, Part 1

Objective 3.3 Given a scenario, research attack vectors and perform application-based attacks

APPLICATION-BASED EXPLOITS

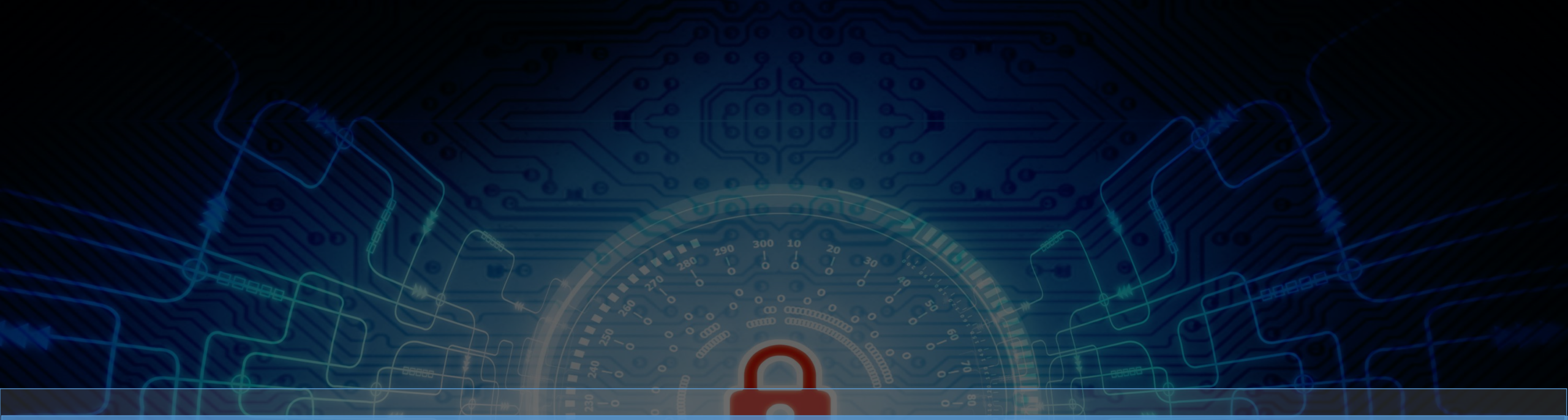
- Injection attack
  - Inserting additional data into application beyond what is expected
  - SQL (Structured Query Language)
    - Adding specially crafted SQL input to extract/modify data or execute commands
  - HTML
    - Adding HTML code/ submitting data to change how a page works or the data is handled

INJECTIONS, cont'd

- Command
  - Adding command line options that change the way commands operate
- Code
  - A generalization of SQL injection – adding code in any language to change a program's behavior

QUICK REVIEW

- Injection attacks provide specially crafted input to applications
- Injection attacks depend on an application's failure to properly validate input data
- Results can include crashing a service or making it unresponsive
- Some injection attacks can provide privilege escalation



Episode 5.03 – SQL Injection Demo

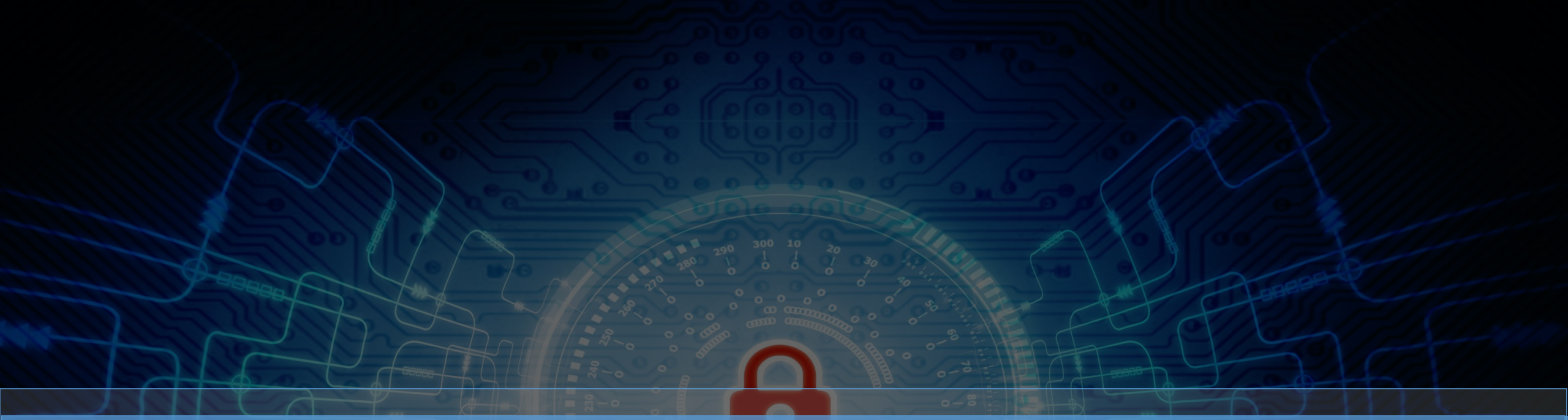
Objective 3.3 Given a scenario, research attack vectors and perform application-based attacks

SQL INJECTION DEMO

- SQL Injection demo

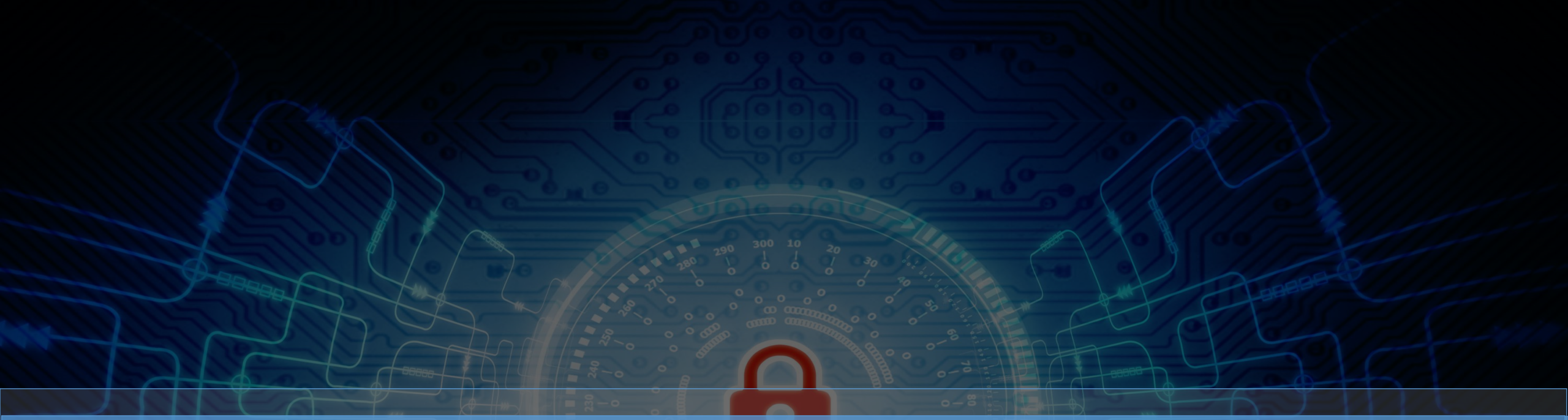
QUICK REVIEW

- Hand crafted SQL injection works in some cases
- Lack of input validation can make any application that uses SQL vulnerable
- sqlmap and metasploit each make SQL injection attacks easy



Episode 5.04 – Labtainers Lab (SQL Injection)

Objective 3.3 Given a scenario, research attack vectors and perform application-based attacks



Episode 5.05 – Application Exploits, Part 2

Objective 3.3 Given a scenario, research attack vectors and perform application-based attacks

AUTHENTICATION EXPLOITS

- Credential brute forcing
  - Offline cracking (Hydra)
- Session hijacking
  - Intercepting and using a session token (generally) to take over a valid distributed (web) session
- Redirect
  - Sending the user to a different site from what they expected (phishing)

AUTHENTICATION EXPLOITS

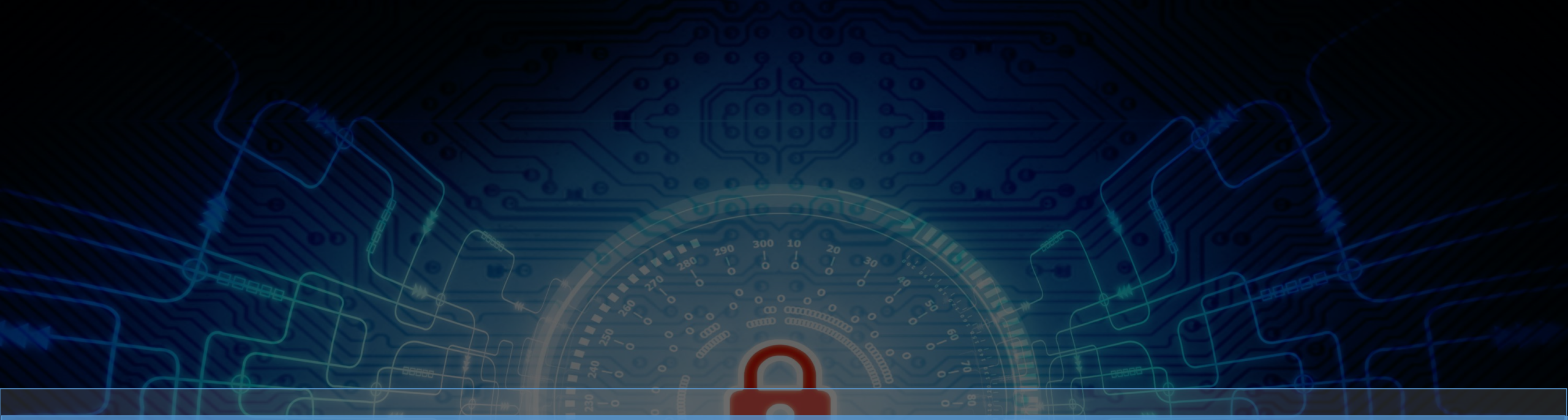
- Default credentials
  - Out of the box artifacts (you have to clean these up!)
- Weak credentials
  - This is why password cracking works
- Kerberos exploits
  - Forged tickets to allow unauthorized access to resources

AUTHORIZATION

- Parameter pollution
  - Providing custom input parameters to alter service/API operation
- Insecure direct object reference
  - Programming mistake that can allow an attacker to bypass access controls and access resources or data

QUICK REVIEW

- Authentication attacks include credential brute forcing, session hijacking, redirecting, and forged Kerberos tickets
- If you can acquire valid authentication credentials, you have access to lots of data
- Authorization attacks include parameter pollution and insecure direct object reference



Episode 5.06 – Application Exploits, Part 3

Objective 3.3 Given a scenario, research attack vectors and perform application-based attacks

CROSS-SITE SCRIPTING (XSS)

- Injection attack in which an attacker sends malicious code (client-side script) to a web application that a subsequent client runs
  - Stored/persistent
    - Attack data (script) stored discretely on the server
  - Reflected
    - Non-persistent attack in which attack code is sent to another client
  - DOM (Document Object Model)
    - XSS attack that uses XML, not HTML, to transport attack code

CROSS-SITE REQUEST FORGERY (CSRF/XSRF)

- Similar to XSS; occurs within an authenticated session
- XSRF attacks a user
- Attacker can cause authorized user to take some action by clicking a link

CLICKJACKING

- Tricking user into clicking a different link or object that was intended
- Attackers can use transparent or opaque layers to embed attack links

SECURITY MISCONFIGURATION

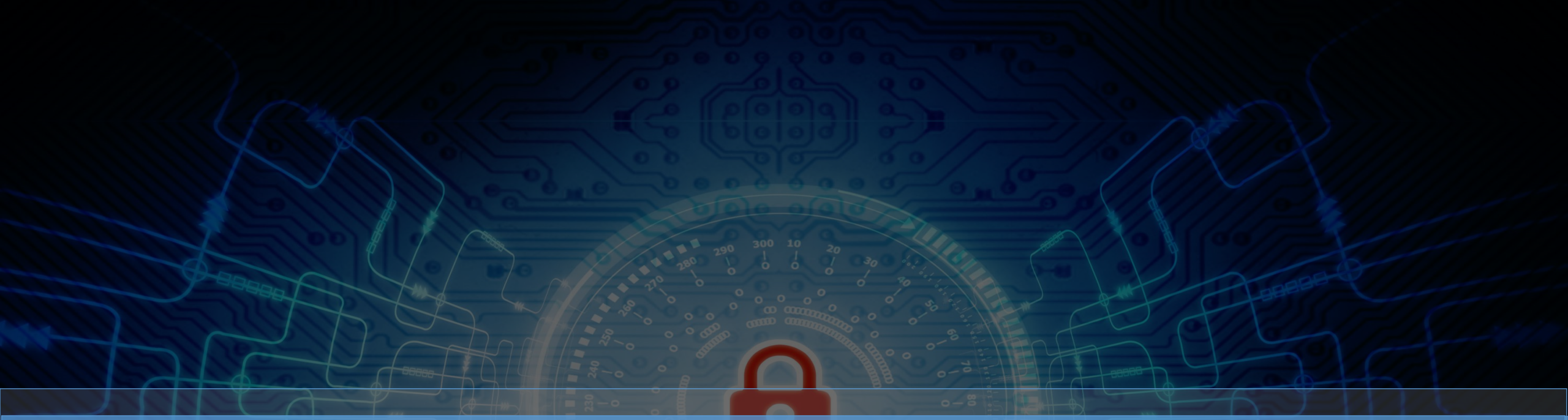
- Directory traversal
  - Allows users to navigate outside a web server's root directory
- Cookie manipulation
  - Access to cookies can allow an attacker to change the way in which a web application operates in general, or just for a specific user/session

FILE INCLUSION

- Related to directory traversal
- Attacker is allowed to build path to .exe file or a file to access
- File can be local or remote

QUICK REVIEW

- XSS is an injection attack on a server using scripting code and has three types: stored/persistent, reflective, or DOM
- XSRF/CSRF attacks the user and occurs within an authenticated session
- XSS and XSRF both use client/server interaction to launch attacks based on specially crafted links or scripts
- Passive attacks exploit security misconfigurations (e.g directory traversal, cookie manipulation, and file inclusion)

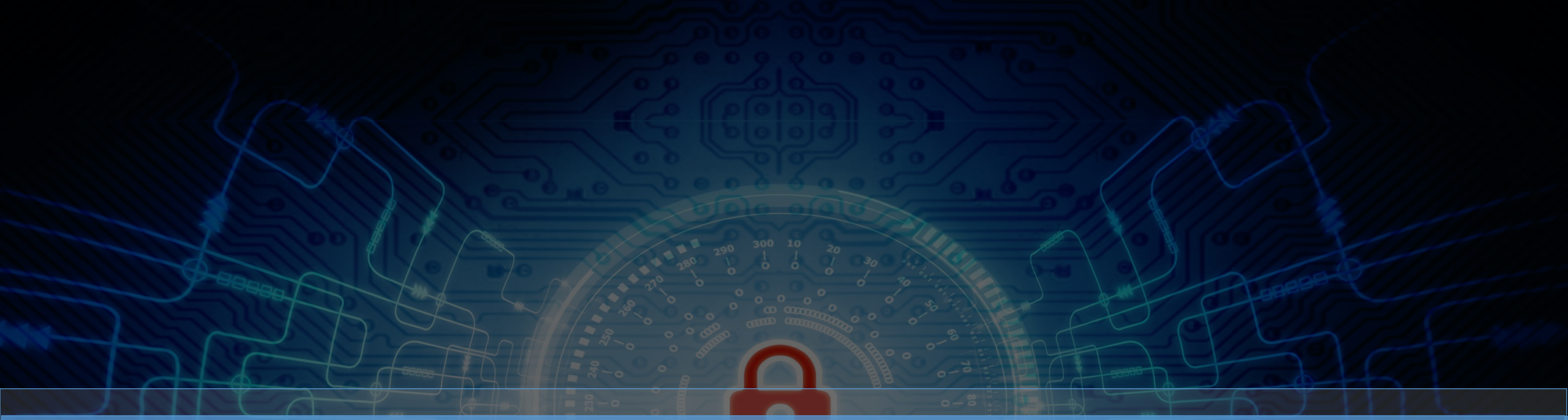


Episode 5.07 – Cross-Site Scripting Demo

Objective 3.3 Given a scenario, research attack vectors and perform application-based attacks

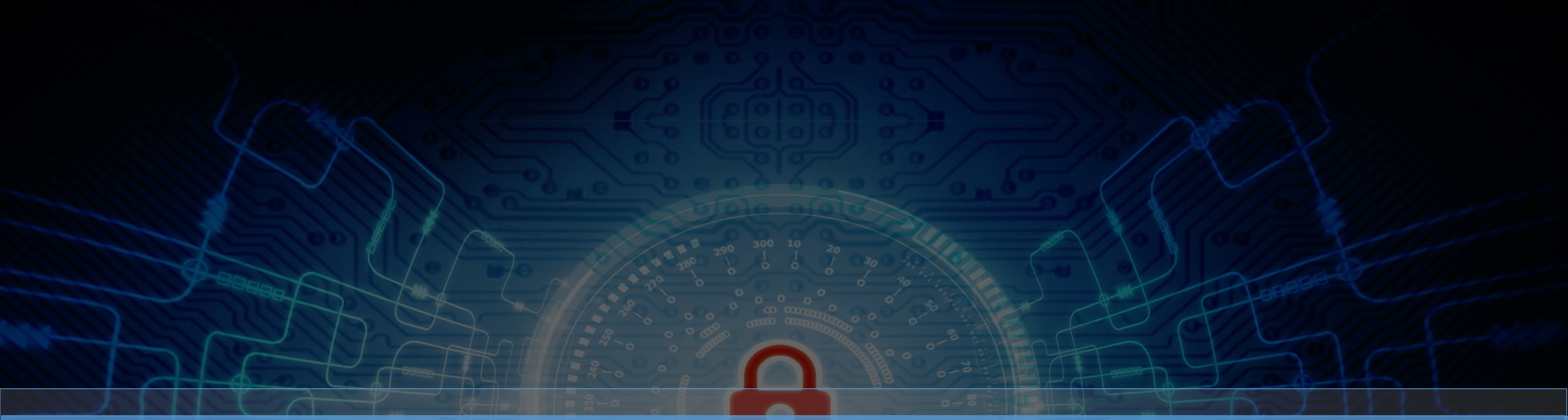
QUICK REVIEW

- XSS can allow an attacker to run almost any script code
- If successful, XSS attacks can compromise many client computers and devices
- Compromise can include remote control, data exfiltration, and setup for further attacks



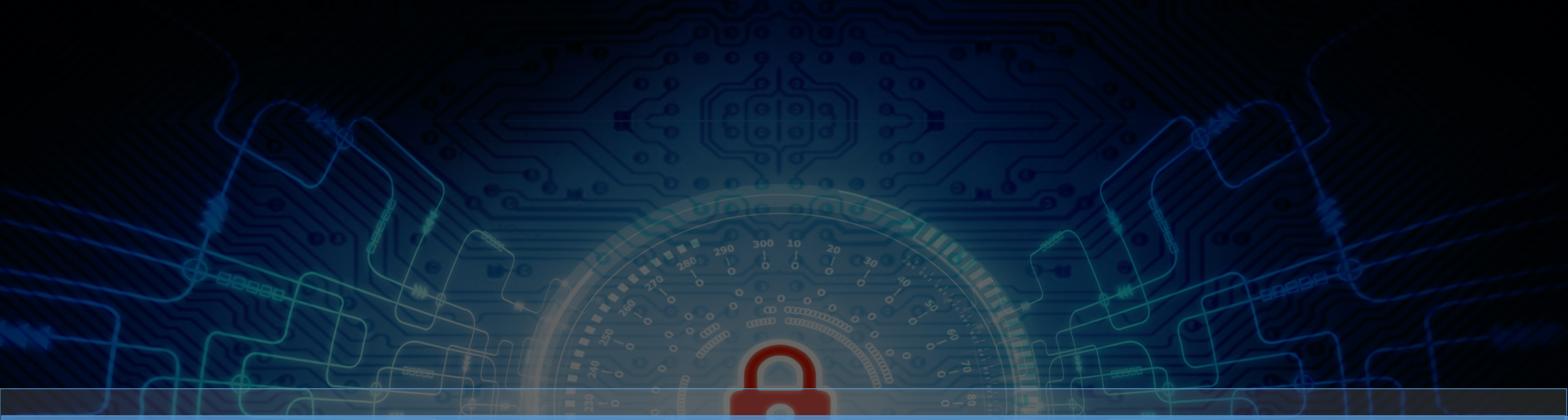
Episode 5.08 – Labtainers Lab (Cross-Site Scripting)

Objective 3.3 Given a scenario, research attack vectors and perform application-based attacks



Episode 5.09 – Labtainers Lab (Cross-Site Request Forgery)

Objective 3.3 Given a scenario, research attack vectors and perform application-based attacks



Episode 5.10 – Code Vulnerabilities

Objective 3.3 Given a scenario, research attack vectors and perform application-based attacks

UNSECURE CODE PRACTICES

- Comments in source code
  - Good for developers and technical personnel
  - Bad for keeping secrets
- Lack of error handling
  - Bad things happen –developers don’t think of everything
  - Unhandled errors can do some cool things

UNSECURE CODE PRACTICES

- Overly verbose error handling
  - Error messages can give too much info
  - Bad error message:
    - “Password invalid for this user”
  - Better error message:
    - “User ID or password is invalid”

UNSECURE CODE PRACTICES

- Hard-coded credentials
  - Happens often – compiled and interpreted (strings command)
  - Attackers can use login credentials
  - Most web apps connect to some other service

UNSECURE CODE PRACTICES

- Race conditions
  - Output is dependent on other event timing
  - Attackers can influence other events and affect operation
  - TOC (Time of Check)/TOU (Time of Use)
    - Checking on some resource, then using it later

UNSECURE CODE PRACTICES

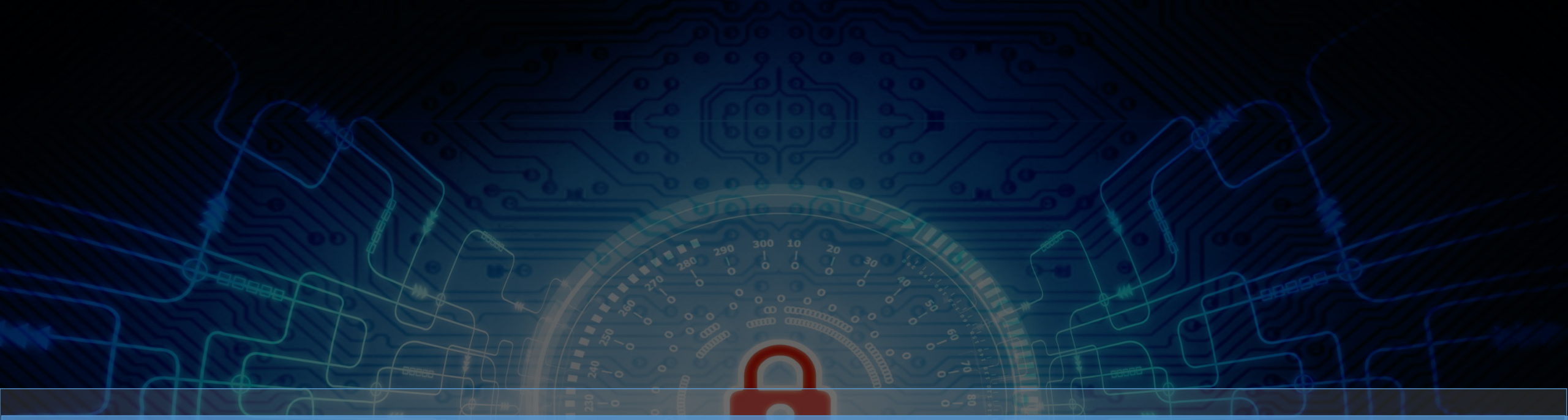
- Unauthorized use of functions/unprotected APIs (Application Programming Interface)
  - Unintended API usage
- Hidden elements
  - HIDDEN attribute in XML and HTML (doesn’t hide data in the source code)
  - Sensitive information in the DOM

UNSECURE CODE PRACTICES

- Code signing
  - Certificates can authenticate author's identity, ensure integrity
- Lack of code signing
  - Lack of signing allows attackers to modify code between deployment and execution

QUICK REVIEW

- Source code comments can provide attackers with valuable insider information
- Error messages can also provide attackers with guidance on how to proceed with an attack
- Any software developer shortcuts (i.e. laziness) can make an attacker's job easier



Episode 5.11 – API Attacks and Attack Resources

Objective 3.3 Given a scenario, research attack vectors and perform application-based attacks

API ATTACKS

- RESTful (Representational State Transfer)
  - Architectural style for distributed applications using HTTP
  - Based on HTTP verbs – simple and lightweight
  - Popular in mobile apps

API ATTACKS

- Extensible Markup Language Remote Procedure Call (XML-RPC)
  - XML used to encode remote procedure calls, transported via HTTP

API ATTACKS

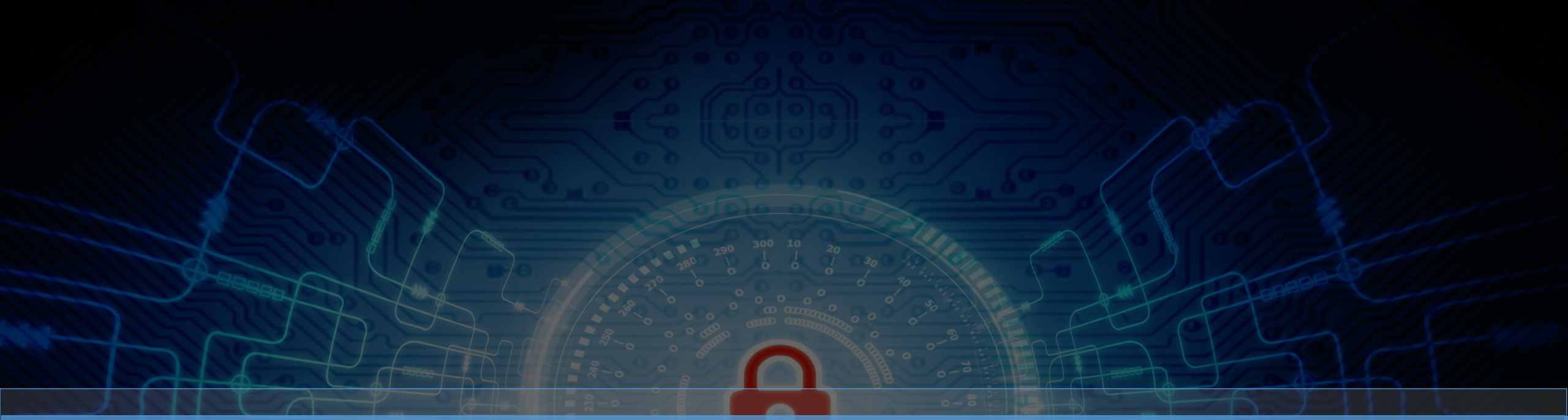
- SOAP (Simple Object Access Protocol)
  - Older, more structured distributed service call architecture
  - Uses payload and envelope

ATTACK RESOURCES

- Word lists
  - Lists of commonly use authentication credentials
  - Cracking utilities use these to carry out brute-force attacks
  - Some tools include common word lists
    - Kali
    - Metasploit framework
    - IKEForce
  - Resources for word lists
    - Wordlistctl
      - https://github.com/BlackArch/wordlistctl
    - WirelessHack.org
      - https://www.wirelesshack.org/wpa-wpa2-word-list-dictionaries.html

QUICK REVIEW

- Distributed computing depends on APIs
- API weaknesses can allow attackers to bypass traditional controls
- Many resources exist online that make attacking APIs easier than building all attacks from scratch



Episode 5.12 – Privilege Escalation (Linux)

Objective 3.3 Given a scenario, research attack vectors and perform application-based attacks

LINUX-SPECIFIC PRIVILEGE ESCALATION

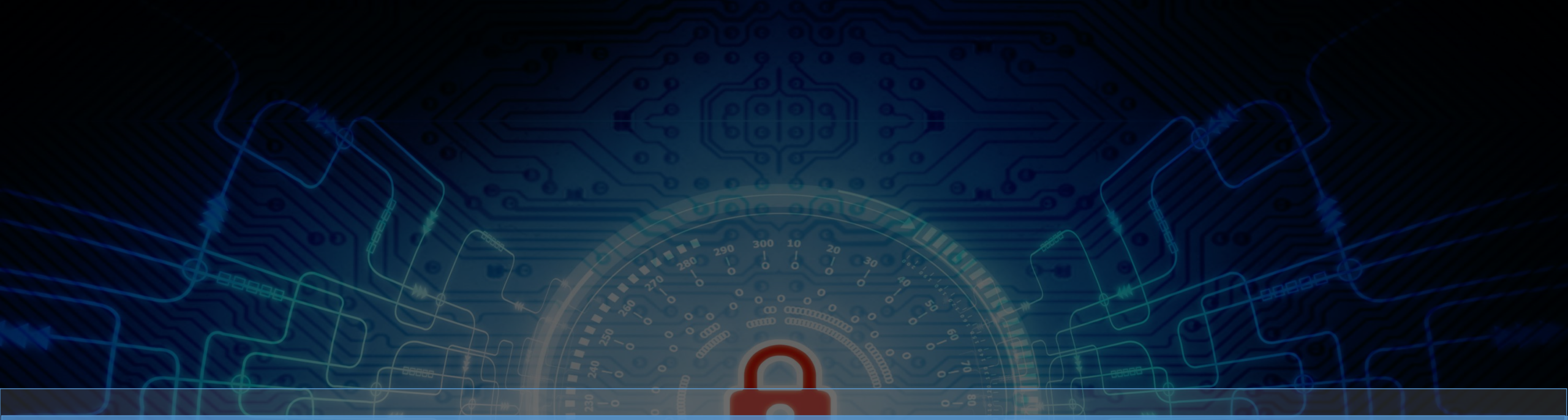
- SUID/SGID programs
  - Permission to execute a program as executable's owner/group
  - *ls -l* shows 's' in executable list of permissions
    - -r-sr-sr-x (SUID and SGID set)
- Unsecure SUDO
  - Authorized users execute commands as if logged in a root

LINUX-SPECIFIC PRIVILEGE ESCALATION

- Ret2libc
  - Stack overflow attack
  - Replaces current stack return address with attacker-chosen address of another subroutine
  - Libc includes useful calls, such as ‘system’
- Sticky bits
  - Directory permission
  - Multiple users can create, read, and write files, but only the owner can delete
  - *ls* shows ‘t’ in the last bit of permissions
    - drwxrwxrwt

QUICK REVIEW

- SUID/SGID and sudo make systems easier to use, but can make them easier to compromise
- Ret2libc is a potential attack vector for hijacking processes
- Sticky bit directories can allow attackers to write files and executables



Episode 5.13 – Privilege Escalation (Windows)

Objective 3.3 Given a scenario, research attack vectors and perform application-based attacks

WINDOWS-SPECIFIC PRIVILEGE ESCALATION

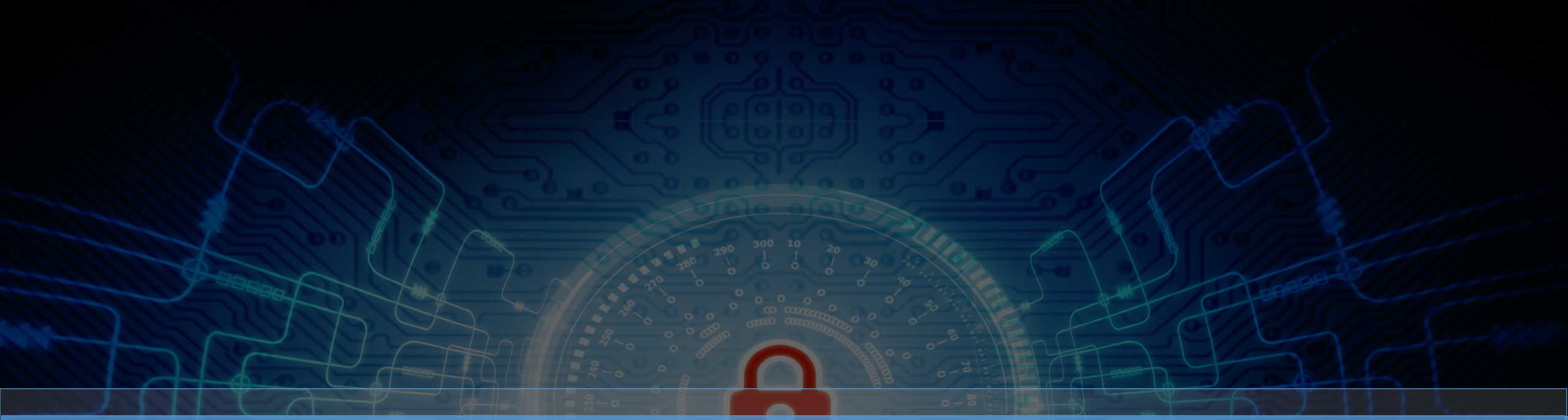
- Cpassword – Group Policy Preference attribute that contains passwords
  - SYSVOL folder of the Domain Controller (encrypted XML)
- Clear text credentials in LDAP (Lightweight Directory Access Protocol)
- Kerberoasting – domain users can query Kerberos tickets for other users
  - https://www.harmj0y.net/blog/powershell/kerberoasting-without-mimikatz/

WINDOWS-SPECIFIC PRIVILEGE ESCALATION

- Credentials in LSASS (Local Security Authority Subsystem Service)
  - Enforces security policy
- Unattended installation
  - PXE (Preboot Execution Environment) credentials
- SAM database (Security Account Manager)
  - Database that contains user passwords
- DLL hijacking (Dynamic Link Library)
  - Forcing a loader to load a malicious DLL

QUICK REVIEW

- Cpassword and LDAP credentials may contain valuable credentials
- PXE (Preboot Execution Environment) credentials can be used to access systems as an authorized user
- DLL hijacking is an attack vector that could allow an attacker to load malware



Episode 5.14 – Misc. Privilege Escalation

Objective 3.3 Given a scenario, research attack vectors and perform application-based attacks

EXPLOITABLE SERVICES

- Unsecure service and protocol configurations
  - Cleartext, legacy options, old protocols, default configuration

EXPLOITABLE SERVICES

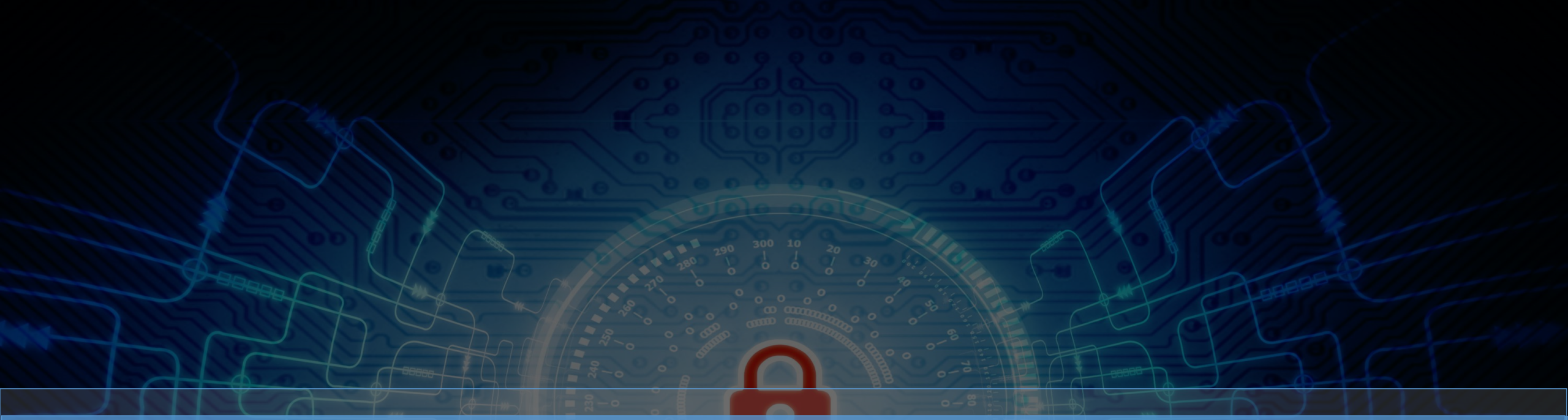
- Unquoted service paths
  - Allow abbreviated attack paths (without spaces)
- Writable services
  - Allow attacker to replace services with malicious programs

PRIVILEGE ESCALATION

- Unsecure file/folder permissions – root installs allow read/write by any user
- Keylogger
  - Records every keystroke
- Scheduled tasks
  - Attacker may add new task to run persistently with elevated privileges
- Kernel exploits
  - Unpatched systems are vulnerable

QUICK REVIEW

- Service path notation without quotes and writable services can allow for service exploits
- Look for files and folders that allow excessive read/write permissions
- Footprinting can provide information on kernel vulnerabilities



Episode 5.15 – Misc. Local Host Vulnerabilities

Objective 3.3 Given a scenario, research attack vectors and perform application-based attacks

LOCAL HOST VULNERABILITIES

- Default account settings – disable accounts that are not being used
- Sandbox escape
  - Shell upgrade – gaining access to a shell with higher privilege
  - VM – escaping a VM may allow access to underlying environment
  - Container – similar to VM escape (i.e. Docker)

PHYSICAL DEVICE SECURITY

- Cold boot attack
  - Ability to physically reboot a system (can allow access to encryption keys)
- JTAG debug (Joint Test Action Group)
  - Can allow attacker to interact with chips
- Serial console
  - If not disabled, provides direct access to servers

QUICK REVIEW

- Default artifacts left in place are almost always vulnerabilities
- A lack of physical security (physical access) always makes attacking easier
- Look for easy attack paths – administrators may have overlooked something