What’s Going On?

- This is a “Capture The Flag” hacking contest
- Teams from a number of Universities/Institutions compete against each other
- Each team has to defend a computer it manages and attack the computers managed by other teams
- The teams have 4 hours for setting up their protections and compromise the other teams’ computers
- A real-time scoring system determines who is the best at defense and attack
Why Is This A Big Deal?

- This is the first time in US history that something like this is attempted
- Other competitions were either
  - Local (e.g., DEFCON in Las Vegas)
  - Limited in the number of teams (e.g., max 3-4)
  - DoD-sponsored (e.g., military exercises)
- This contest involves graduate students, it’s completely “open”, and has a rigorous scoring system
- This competition includes 14 teams from Universities and Institutions spread across the nation
  - UCSB (4)
  - Georgia Tech (3)
  - United States Military Academy (3)
  - University of Texas at Austin (1)
  - Naval Postgraduate School of Monterey (1)
  - University of Illinois, Urbana-Champaign (1)
  - North Carolina State University (1)
How Does This Work?

Virtual Private Network

Wujing

Ishanshade

Shrek

Buzz

UT_Comsoc

0x90

Open kiosk

Jmp esp

Generic Tigers

NC STATE UNIVERSITY
Teams’ Hosts

- OS image configured with a number of services running on VMWare
  - Red Hat 9.0 on VMWare 4.0.5 on Red Hat 9.0
- Service examples
  - World Wide Web, FTP, Audio streaming, Custom services
- Services have a number of exploitable vulnerabilities
- Each OS image/service set is customized to a particular team
- OS images are distributed at the beginning of the day
- Source code for some services is distributed at the beginning of the actual contest
Flags

- Each service has one or more flags associated with it
- When a service is (re)started the flags are initialized to the initial values for the hosting team
- The initial value for service X running on the OS of team A is different from the initial value of service X running on the OS of team B
- To own another team’s service
  - Determine the initial value for the service flag for your team
  - Write the initial value into the other team’s flag for that same service
Monitoring/Scoring

• Each service is equipped with
  – A “get flag” method
  – A “set flag” method
• Getting and setting the flags do not involve exploiting a vulnerability
• A service can be in different states
  – Dead
  – Running
  – Functional (running and flags can be retrieved and set)
  – Functional and Owned by the hosting team
  – Functional and Owned by another team
Monitoring/Scoring

- The scoring systems attempt to read the flag
  - If no connection can be established, the service is considered down
    - No points are assigned
  - If the flags are not accessible, the service is considered non-functional
    - No points are assigned
Monitoring/Scoring

- The flag is analyzed
  - If the flag is the initial flag value of the hosting team
    - A new hash chain for the service is initialized
    - The team receives no points
  - If the flag is the value of the hosting team and the number of get/set iterations in the hash chain is greater than a threshold (e.g., 3)
    - A new flag value is written
    - The hosting team receives points
  - If the flag is the initial value of another team
    - A new flag value is written and a new hash chain is started
    - The other team immediately receives points
  - If the flag is the correct value of the hash chain of another team
    - The other team receives points
Example: Normal operation

• Service X is started on the OS of team A
  – X’s flag is automatically set to $f_0 = c9a56d2822463b$
• The scoring system reads the flag and starts a new hash chain $c_1$ for service X owned by A on host A
  – X’s flag is set to $f_1 = \text{hash}(c_1, X, A, A, f_0)$
• Periodically, the scoring system reads the flag, checks its value against the last value stored for the current hash chain, and the values match
  – X’s flag is set to $f_2 = \text{hash}(c_1, X, A, A, f_1)$
• This operation is repeated a number of times (e.g., 3) before the hosting team starts acquiring points
Something went wrong

- If the flag does not contain the value that was set by the scoring mechanism during the last iteration, several things may have happened:
  - The service has been restarted by the hosting team
  - Another team put their flag for the service
  - Some garbage got written on the flag value
Example: Service is restarted

- If a service is restarted the flag is reset to the initial value f0
- The scoring mechanism starts a new chain c2 for service X owned by team A on host A
- The scoring mechanism writes a new flag value f1=hash(c2,X,A,A,f0)
- Points will be assigned after a number of iterations (e.g., 3)
Example: Service is Owned

- The new flag is the initial flag for the service associated with another team (say g0 = 528668d2e22fa)
- A new hash chain c3 is started and the flag is set to f1(c3,X,B,A,g0)
- The service is owned by the team and points are assigned to that team immediately
- Note: there is no way to know if one of your service is owned by another team by just looking at the flag value
Example: Flag is corrupted

- The flag does not match the value in the current hash chain and also does not match the initial value for any of the teams
- The service is considered non-functional and no points are assigned
- Note: this condition can be detected by looking at the scoring panel
Scoring Panel

- The scoring panel provides a snapshot of the status of the CTF
- It is accessible through a web page (refreshed every 30 seconds)
- It provides information of team’s ping connectivity (ability to answer to ping probes)
- It provides information about the status of services
  - Down
  - Running but non-functional
  - Functional
  - Note: It does not provide information about the ownership of a service
Scoring Panel

• It provides information about how many services are owned by a team
  – This is useful if a team wants to check if the attack was successful because the number of services owned will increase

• It provides information about the performance of a team in the last scoring period (say, last 10 minutes)

• Note: It does not provide absolute score values

• Penalties to the scoring value can be assigned because of improper behavior (e.g., DOS attacks)

• The final winner will be declared only at the end of the exercise
Attack Techniques

- Buffer overflow
- Format string
- Shell attacks
- Race conditions
- Misconfigurations
- Authentication attacks
- Web-based attacks
  - Directory traversal
  - Cookie-based services
  - Cross-site scripting
  - Server-side applications
    - Lack of parameter validation (e.g., SQL injection)
Skills

• Scanning
• Firewalling
• For each type of vulnerability
  – How to identify a vulnerability
  – How to exploit a vulnerability
  – How to patch a vulnerability (without disrupting the get/set flag methods)
  – How to detect a vulnerability
• For each service
  – How to monitor the requests to a service
  – How to monitor the execution of a request
  – Protocol security analysis
  – Application security analysis