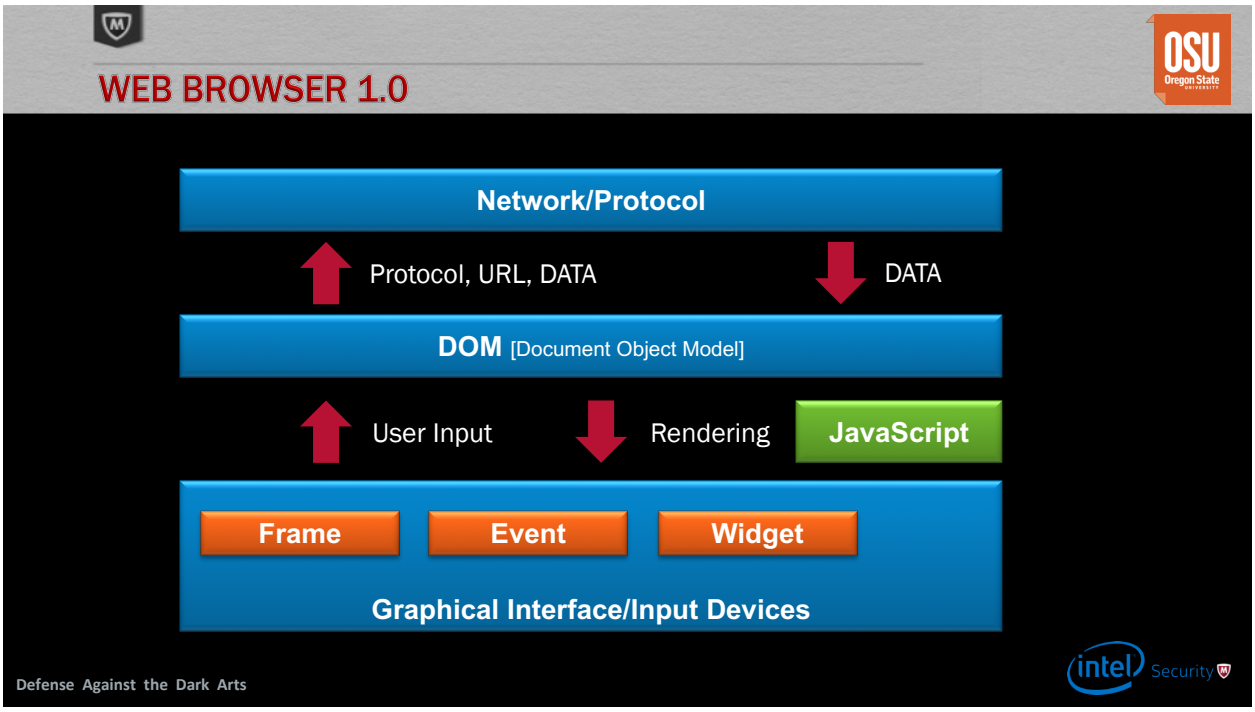
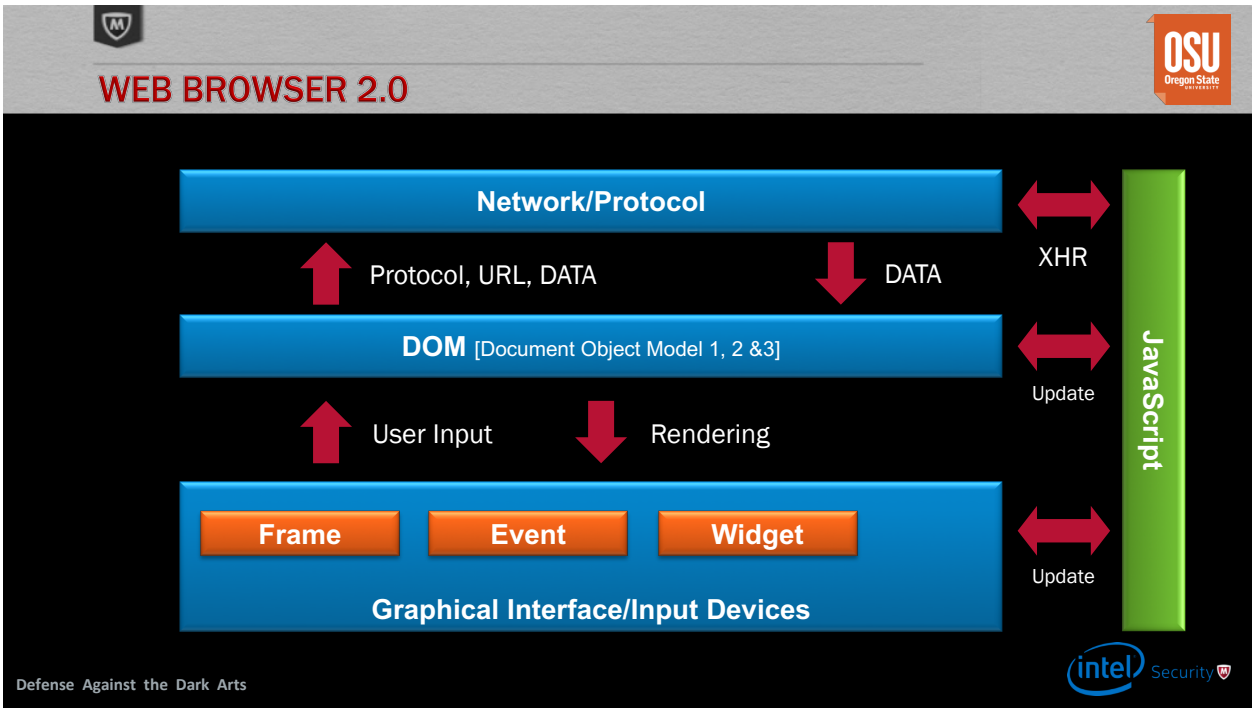
Johnny Zhong
CS373
Web Security

Web Delivery Mechanism

- Browsers change often and security is updated quickly.
- Evolution of malware delivery: Phishing, sql injections/browser exploits, sophisticated phishing/plugin attacks, attacks in browser/spearfishing
- 95% of all malware is delivery through the web



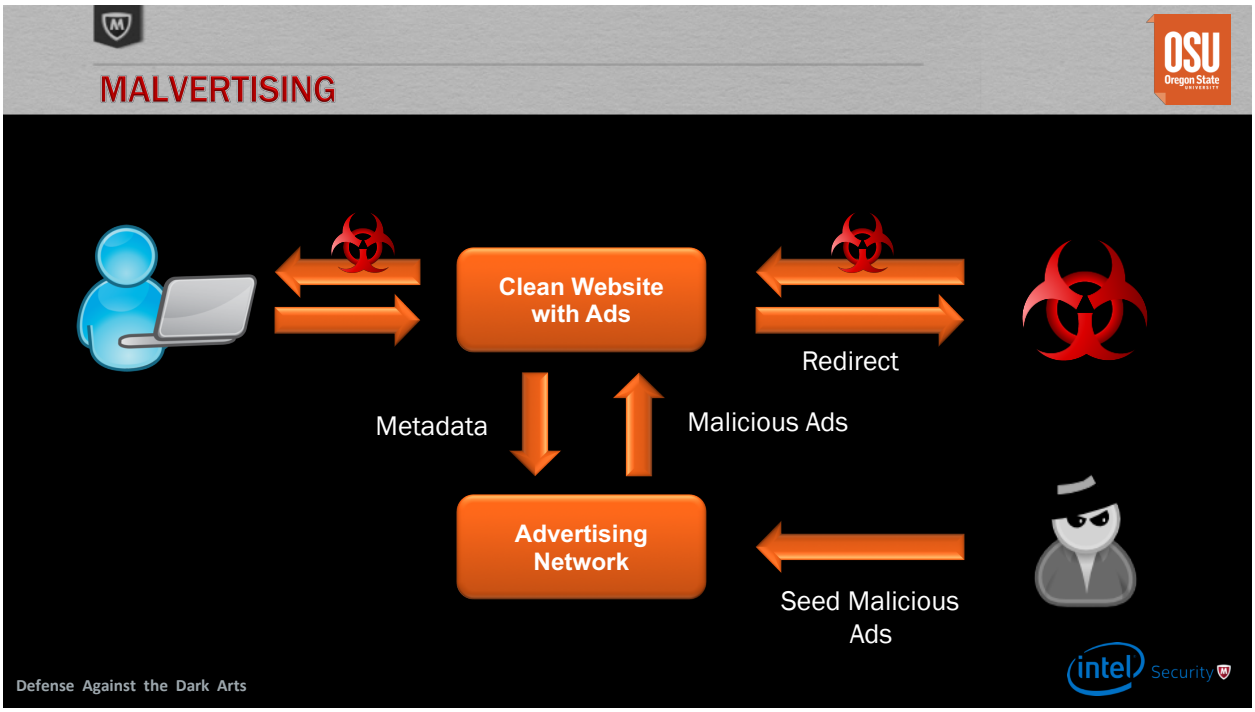


Changes to the browser allow malware attackers to stay in the browser and access the OS through JS.
Malware attackers could hook into Windows APIs through the browser, allowing them access to all traffic between the OS and the browser.

User Level Attacks

- Commonly performed through social engineering
- Users are the weak link
    - Hardening browsers is useless
    - Impatient
    - Lazy
- Phishing
    - Fake websites, similar URLs, signed certificate
    - Abusing user trust
- SEO Poisoning
    - Popular search terms are common malware vectors
    - Developer forums
    - Image searches
- Fake AV/Fake Updates
    - Preys on user's good intentions
    - Popups that look like AV with a full support crew
- WYSIWYG
    - Fake URLs that look like the actual website (rnicrosoft vs microsoft)
    - Toolbars!
- Social Media Attack

- Catfishing
- Fake profiles



- Malvertising
  - Targets: CXOs in medical fields
  - Seed advertising network with ads specific to the target
  - Ads go through the clean website and the ad is clicked by the target
- Waterhole
  - Hack 1 location that will attract your target
  - Developer forums
  - Hack the forum
  - Deliver malicious payload
- Common Defenses
  - Reputation
  - Site Certification
  - BIG RED SIGNS
  - PCI Compliance
  - Conventional AV
  - URL Shortener
  - HUMANS
  - Search authentication

Browser Level Attacks

- Many attacks are multi-layered, due to new security techniques present now
- MitM, MitB
- Modern Browsers – like mini operating systems

- Exploits:
    - Typically requires users to interact
    - Can't filter malicious code due to interpreted nature of JS
    - JS has visibility to your system
    - JS also has interaction with many different levels of the browser
    - Can be obscured because other parts of the browser can provide encoded portions of the script (like a big puzzle)
    - Packer code might be used by clean websites (CNN, for example)
    - Obfuscated code can also be used to prevent people from copying objects/data
    - MitM attack
        - Free WiFi
        - A lot of content is unencrypted
        - Installation of malware through hidden iframes
    - MitB attack
        - No need for malware to leave the browser memory
        - No executable
    - Bad USB
        - Can be a NIC
        - OS detects as NIC, can retrieve network information or redirect DNS
    - Clickjacking – UI redressing
        - Change what the user clicked by overlaying an interface over the real interface
    - SQL Injection
        - Provide sql through a post
        - Mitigated by sanitizing inputs
    - Cross-Site Scripting
        - Stored on forum or through a link
            - Ask server via JS to do something with cookies or credentials
            - Can send details to somewhere else (attacker)
        - Move users to desired locations
    - Non Persistent XSS
        - All information is in URL
    - Cross Site Forgery
        - Exploit server trust in browser
        - Perform some action on server on behalf of server
        - Use of GET request
-
-

WebGoat:

- Free for all attack target
- Fake target
- Not a perfect replica

Webgoat Notes

- Start Firefox, pull the XPI file in order to install.
- Modify .sh file to remove check for Java.
- Change permissions (777)
- Execute webgoat as root (sudo) – sudo sh ./webgoat start 8080

Web Security

- SQL Injection
    - Play with parameters in the URL
    - Allow db to compute the input to determine nature of the website
    - Use math (2-1) to see what response you get. If 1 and redirects to an expected location, then we're interacting with the db. Otherwise, something else might be reading the input and deciding what to do with it.
    - MySQL automatically casts inputs
    - Can exploit database interface by writing valid sql
    - Fingerprint the database vendor by using a script

Webgoat

- Stored XSS
    - Stored – want to get a value on the database behind the website.
    - <script> code here </script>
- Use of Inspect (chrome) and Tamper tools (firefox)
    - To change inputs and requests

Session 2 - Methods to classify/score URLS

- Contextual Data
- Alexa
    - How prevalent the domain is
    - Very prevalent, generally clean. Not prevalent for a long period of time, generally clean
    - Sometimes dirty samples are prevalent for a short period of time, then die down to irrelevancy.
- Archive.org
    - Might be used by researchers to view a defaced website
    - Need some kind of archive for a report
- IPVoid
    - Large list of URLs

    - IPs are being recycled, bad guys commonly reuse the IP
    - IPs are hard to get
    - Normal websites generally have multiple IPs
    - Bad domains generally have only 1 IP
    - Bad domains also won't necessarily have a mail server (so that you can't contact them)
    - Fastflux – behavior of malware to change IPs. They generally have IPs that belong to consumers.
        - Consumer IPs as opposed to business IPs
        - Need to talk back to the home (sinkholing)
- CheckShortUrl
    - Could check where the url goes to
    - Could be completely different
- Site Dossier
    - Give a bigger picture around what the domain is
    - Provides what points to the website
    - How did a user get to the website
- Webutation
    - URL Reputaton Clearinghouse
- Web Inspector
    - Input a URL, does some info
    - Many websites that check scripts for you
- Virus Total
    - Has information about URL files
    - Aggregator of malware and bad URLs
- Linus JWhois
    - Domain registration data client
    - Commonly, bad domains use the same registration information
- Linux Dig
    - DNS Resolver Utility
    - For a good/common domain, they'll have a lot on their DNS (mail server, multiple servers, etc.)
    - For bad domains, they'll generally just have 1 webserver.
    - In json file provided.
- IOC
    - Indicator of Compromise
    - Some kind of detector using indications left by malware

Research Tools

PhantomJS - Assists researchers by automatically render content and look for malware/exploit
JSUNPACK

- detects exploits that target browser and browser plug-in vulnerabilities
- Static file scanning

Burp Suite

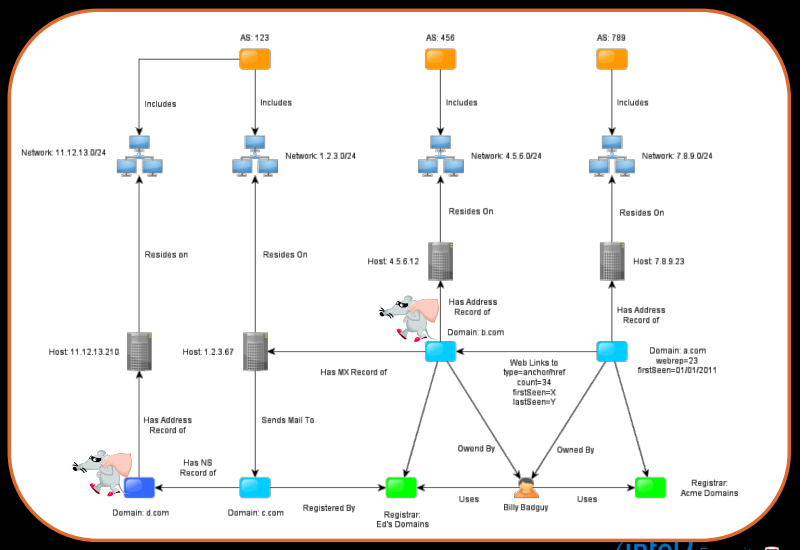
- Intercept and modify request content
- Spidering capability

Webscarab – similar to burp suite

Intelligent Methods of Classifying URLs

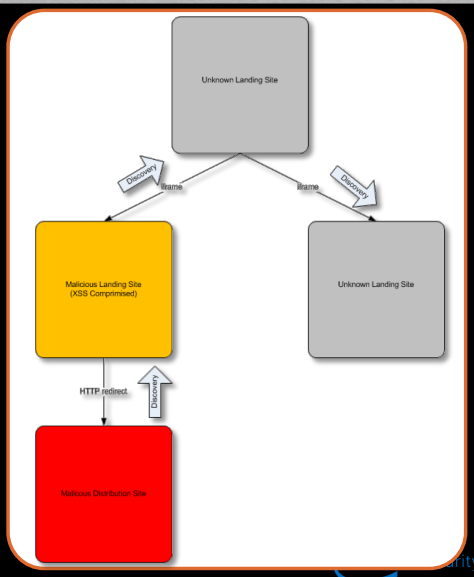
GRAPH-BASED URL CLASSIFICATION

- New graph database engines are filling a relevant need.
  - Examples
    - Neo4J
    - Titan
- Ability to traverse a directed graph quickly and efficiently
- Ability to store very large data sets
- Classify URLs by "guilt by association"



GRAPH-BASED URL CLASSIFICATION - EXAMPLE

- We know about malicious landing site and malicious distribution site.
- Traverse the graph to find other inbound and outbound links to new entities
- Requires Automated crawling and link/feature storage



THE EXPERT SYSTEM

- Too many URLs to hand classify, or even crawl automatically.
- Modern research systems classify most URLs ***without*** ever seeing content
- Layer the approach from light to heavy
- You may be asked to classify millions of URLs per day
- How would **YOU** do it?