Lazily Finding Holes Without Breaking The Law

Nick von Dadelszen – Technical Director
Event – OWASP Day 2015
Date – 27th February 2015
Company Overview

Lateral Security (IT) Services Limited

Founded in April 2008 by Nick von Dadelszen and Ratu Mason (Both Directors)

Auckland, Wellington, Melbourne: ~20 highly specialised security consultants

Services
- Security testing (design & architecture, penetration testing, configuration, code reviews, security devices & controls, mobile apps)
- Security advisory (Lifecycle compliance & audit - ISO, PCI-DSS, NZISM, policy, process development, threat modeling and risk assessment)
- Regular ongoing technical testing and assurance programs
The Catalyst
The Catalyst

- When scoping jobs I use browse through Burp to get a site map and to log my activities
- Potential client rings up asking for quote
- I browse site to assess functionality
- After scoped, I casually check Burp and notice a passive scan alert
- Turns out the site has some creds in JavaScript which turns into a fairly big deal
- Have to responsibly disclosure a serious security issue during the sales process
The Bright Idea
The Bright Idea

- What if I could be looking for vulnerabilities all the time when I browse the internet
- What if I could find them without even trying
- I could pump all of my browsing traffic through a proxy that searches for vulnerabilities passively
- Must not break the law and perform active attacks
The Law

I AM THE LAW!
Accessing computer system without authorisation

(1) Every one is liable to imprisonment for a term not exceeding 2 years who intentionally accesses, directly or indirectly, any computer system without authorisation, knowing that he or she is not authorised to access that computer system, or being reckless as to whether or not he or she is authorised to access that computer system.

(2) To avoid doubt, subsection (1) does not apply if a person who is authorised to access a computer system accesses that computer system for a purpose other than the one for which that person was given access.
The Implementation

Trying is the first step towards failure.
The Implementation
The Implementation

- I am lazy

- Burp suite already has a passive scan engine so let's just use that

- Want to run it in headless mode on a remote server so I can use it wherever I am without having to load Burp etc

- Should log all passive scan results to console and a file

- Should capture full packets for any high severity issue identified
The Implementation
The Problems

MY CODE DOESN'T WORK AND I DON'T KNOW WHY

MY CODE WORKS AND I DON'T KNOW WHY

http://wanna-joke.com
The Problems

- Adding Burp extensions in headless mode is hard
  - Had to call on the skills of Feabell to solve this one
  - Used a bootstrap extender to load a saved config then load our main extension
- Java is crap and keeps crashing
  - Needed to restart every 1000 issues
Adding Custom Scan Items

**Python**

This is plagiarism. You can't just "import essay."

**Java**

I'm two pages in and I still have no idea what you're saying.
Adding Custom Scan Items

- It's easy to add custom scans to Burp

- It is easier to add custom scan items to Burp using Python than Java

- Have I ever told anyone that I have Java?

- Have to work out how to load Python extensions in headless mode
def doPassiveScan(self, baseRequestResponse):
    #print "Starting doPassiveScan..."
    analyzedResponse = self.helpers.analyzeResponse(baseRequestResponse.getResponse()) # R
    analyzedRequest = self.helpers.analyzeRequest(baseRequestResponse)

    #print 'test'
    url = analyzedRequest.getUrl()
    params = analyzedRequest.getParameters()

    issues = list()

    # url checks
    #print url.getPath()

    # Padding Oracle
    if 'WebResource.axd' in url.getPath() or 'ScriptResource.axd' in url.getPath():
        if PaddingOracleCheck(params):
            print "Found Padding Oracle issue"
            issues.append(PaddingOracleIssue(baseRequestResponse.getHttpService(),
                                              analyzedRequest.getUrl()))

    #DependencyHandler
    if 'DependencyHandler.axd' in url.getPath():
        if DependencyHandlerCheck(params):
            print "Found DependencyHandler issue"
            issues.append(DependencyHandlerIssue(baseRequestResponse.getHttpService(),
                                                  analyzedRequest.getUrl()))

    if len(issues) > 0:
        return issues
    else:
        return None
Padding Oracle
Padding Oracle For Dummies

- Crypto is hard!!

- Sometimes the end-user is used as a mechanism to transfer encrypted data

- This allows the end user to perform crypto attacks against the mechanism

- Padding Oracle occurs when a user can modify the encrypted text to obtain at least three different results:
  - Cipher text gets decrypted, resulting data is correct.
  - Cipher text gets decrypted, resulting data is garbled and causes some exception or error handling in the application logic.
  - Cipher text decryption fails due to padding errors.
Padding Oracle For Dummies

- Padding Oracles found in multiple frameworks and applications:
  - ASP.Net
  - JavaServer Faces
  - Ruby on Rails
  - SAML
  - SSL
    - Lucky Thirteen
    - POODLE

- Can allow both decryption and encryption of messages

- The fix is to append a HMAC to the encrypted message to prevent message tampering
Testing For Padding Oracle Passively

- The ASP.Net patch (MS10-070) appended an HMAC to encrypted payloads

- This can be checked easily by reviewing requests to WebResource.axd and ScriptResource.axd

- A previously released tool (ms10-070_check.py) already does this

- I re-implemented this check as a passive vulnerability check in Burp

- Can now detect ASP.Net Padding Oracle just by browsing sites
The Results

KEEP CALM AND WAIT FOR THE RESULTS
The Results

• Ran on and off for a couple of months
  • Over 24,000 issues logged

• Majority were boring:
  • Cleartext passwords
  • Cookie issues
  • Session token in URL

• Several more interesting:
  • Padding Oracle
  • SQL statement in request parameter

• Had to responsibly disclosure one issue during the exercise
What Is With .mil?

MILITARY INTELLIGENCE
It sounds way better on paper.
What Now?
What Now?

• So, where to from here?
• Should I publish my proxy publicly?
• Set up a TOR exit node?
• Want to re-implement standalone rather than relying on Burp
Questions and Contacts

Lateral Security (IT) Services Limited

Wellington
38-42 Waring Taylor Street (level 7, Petherick Tower)
PO Box 8093, Wellington 6143, New Zealand
Phone: +64 4 4999 756
Email: sas@lateralsecurity.com

Auckland
187 Queen Street (level 8, Landmark House)
PO Box 7706, Auckland, New Zealand
Phone: +64 9 3770 700
Email: sas@lateralsecurity.com

Melbourne
200 Queen Street (level 13)
Melbourne, VIC 3000, Australia
Phone: +61 1300 554745
Email: sas@lateralsecurity.com

Presentation Download
www.lateralsecurity.com/resources.html