Automation of Web Application Scanning with Burp Suite

Abakumov Andrey, Zaitov Eldar
whoami

> Andrey Abakumov

> Yandex Product Security Team

> BugBounty (Uber, Facebook, Qiwi and others)

> CTF player
Architecture

Web UI

REST API

Celery

Agent 1

Agent 2

Agent 3
Architecture of agents

Crawler → Entry point → Message handler → Active scan → Passive scan → Report
Why Burp Suite?

Burp Suite - GUI proxy server for manual analyse of HTTP and Websocket protocols

› Everyone in our team uses Burp Suite
› The ability to write your own plugins
› Large open source community
› New approaches, researches from the creators of the product
Burp Suite

Proxy
Repeater
Spider
Scanner
Extender
Intruder
...

Active
Passive
Burp Suite

› Agent in scanning platform
› Running in headless/console mode
› Loads 2 plugins:
   Automation of the scanning process;
   The collection of active and passive checks.
How to create simple Plugin
Burp Extender Api

Java Interfaces with Javadoc

› Java
› Python
› Ruby
package burp;

public class BurpExtender implements IBurpExtender {
    @Override
    public void registerExtenderCallbacks(IBurpExtenderCallbacks callbacks) {
        /* Code of plugin */
    }
}
public class BurpExtender implements IBurpExtender, IExtensionStateListener {
    @Override
    public void registerExtenderCallbacks(IBurpExtenderCallbacks callbacks) {
        callbacks.registerExtensionStateListener(this);
    }

    @Override
    public void extensionUnloaded() {
        /* Run code when Unload Extension*/
    }
}
public class BurpExtender implements IBurpExtender, IHttpListener {
    @Override
    public void registerExtenderCallbacks(IBurpExtenderCallbacks callbacks) {
        callbacks.registerHttpListener(this);
    }

    @Override
    public void processHttpMessage(int toolFlag, boolean messageIsRequest, IHttpRequestResponse messageInfo) {
        /* Read and change any request/response */
    }
}
Usefull methods

IExtensionHelper
- IRequestInfo analyzeRequest(byte[] request);
- IResponseInfo analyzeResponse(byte[] response);
- IParameter getRequestParameter(byte[] request, String parameterName);
- byte[] addParameter(byte[] request, IParameter parameter);
- byte[] toggleRequestMethod(byte[] request);
- byte[] buildHttpMessage(List headers, byte[] body);

IBurpExtenderCallbacks
- IHttpRequestResponse makeHttpRequest(IHttpService httpService, byte[] request);

IRequestInfo
- List getHeaders();
Burp-molly-scanner
Plugin for automation

› Reads configs
› Accepts requests
› Implements a business logic (authentication etc.)
› Deduplicates Request/Response
› Invokes Burp Suite Active Scan
› Waits for active scans to complete or timeout
› Generates issues report in XML format
Accepts requests

› Accepts request from Crawler, because Burp suite can’t render Javascript
› Runs Burp Suite Spider callbacks.sendToSpider(extConfig.getInitialURL());

V1.0

› Application in Golang
› Runs PhantomJS with Burp Suite as Proxy server
› Renders Javascript
Process all requests

```java
public void processHttpMessage(int toolFlag, boolean messageIsRequest, IHttpRequestResponse messageInfo)

› Modify User-agent
› Add custom Parameters
› Do custom authentication
› Run Active and Passive scans
```
Problem of scanning large services
Problem of scanning large services
Deduplication

- Deduplication by Murmur hash of Response

Response1 → Hash1 → Bloom Filter → Active scan

Response2 → Hash2 → Present in bloom filter?

No
Deduplication

Deduplication by Simhash

Response1

HTML?

Yes

Parse HTML

Simhash1

No

Simhash1

Response2

...

Simhash2

Tree

Active scan

Distance > n?

Finding distance between hashes

Yes

Yes
Waits for all active scans

```java
scanners = Collections.synchronizedList(new ArrayList<IScanQueueItem>());
Iterator<IScanQueueItem> i = scanners.iterator();
while (i.hasNext()) {
    IScanQueueItem scan = i.next();
    if (scan.getStatus().equals("finished")) {
        i.remove(); ...
    }
```
Burp-molly-pack
public class BurpExtender implements IBurpExtender, IScannerCheck {
    @Override
    public void registerExtenderCallbacks(IBurpExtenderCallbacks callbacks) {
        callbacks.registerScannerCheck(this);
    }
    @Override
    public List<IScanIssue> doActiveScan(IHttpRequestResponse baseRequestResponse, IScannerInsertionPoint insertionPoint) {
        /* Code for active scanning */
    }
}
@Override
public List<IScanIssue> doPassiveScan(IHttpRequestResponse baseRequestResponse{
    /* Code for passive scanning */
}
@Override
public int consolidateDuplicateIssues(IScanIssue existingIssue, IScanIssue newIssue) {
    /* Deduplicate issues */
}
public List<IScanIssue> doScan(IHttpRequestResponse baseRequestResponse, IScannerInsertionPoint insertionPoint) {
    ...
    for (String payload : CRLFSplitters) {
        IHttpRequestResponse attack = this.callbacks.makeHttpRequest(httpService, insertionPoint.buildRequest(this.helpers.stringToBytes(payload)));
        IScanIssue res = analyzeResponse(attack, insertionPoint, payload);
        if (res != null) issues.add(res);
    }
    ...
}
A collection of security checks

- XSS
- XXE
- SQLi
- CSRF
- ... 
- SSTI
- Disclosure
- Cookie/Headers
- Insecure CORS
- Command Injections
- JSONP
- SSRF
- CRLF
- Websockets
- Express Redirects
- ... 
- XXE
- HTTPoxy
- Java Deserialization
- CSP
Websocket Plugin

Request

Search Websockets Header

Search and delete Origin Header

Yes

New security issue

Http code 101?

Add header
Origin: https://evil.com
SSRF, XXE ...
Manual search of vulnerabilities

<table>
<thead>
<tr>
<th>Host</th>
<th>Method</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>htt</td>
<td>POST</td>
<td>/api/v2</td>
</tr>
<tr>
<td>htt</td>
<td>POST</td>
<td>/api/v2</td>
</tr>
<tr>
<td>htt</td>
<td>POST</td>
<td>/api/v2</td>
</tr>
<tr>
<td>htt</td>
<td>POST</td>
<td>/api/v2</td>
</tr>
<tr>
<td>htt</td>
<td>GET</td>
<td>/api/v2</td>
</tr>
<tr>
<td>htt</td>
<td>GET</td>
<td>/api/v4</td>
</tr>
<tr>
<td>htt</td>
<td>GET</td>
<td>/api/v4</td>
</tr>
<tr>
<td>htt</td>
<td>GET</td>
<td>/api/v4</td>
</tr>
<tr>
<td>htt</td>
<td>GET</td>
<td>/api/v4</td>
</tr>
<tr>
<td>htt</td>
<td>GET</td>
<td>/api/v4</td>
</tr>
</tbody>
</table>

Issues

- External service interaction (DNS)
- External service interaction (HTTP)
- SOME (Same Origin Method Execution) [6]
- SSRF Molly HTTP Interaction [24]
- SSRF Molly DNS Interaction [48]
- ! Interesting input handling: Magic value: null [13]
- Cross-origin resource sharing: unencrypted origin trusted
- Cross-origin resource sharing: all subdomains trusted
- i Cross-site scripting (reflected) [21]
- i Cross-origin resource sharing
- i Robots.txt file
- i Cacheable HTTPS response [18]
- i Clickjacking [6]
- i Missing X-Content-Type-Options header [6]
Integration with Bugbounty

Bugbounty -> Information Security Engineer -> New scan module

Report -> Rescan all services
Limitations

› You need Crawler to render Javascript

› You can’t create your own Type index - Extension generated issue

› You can’t create global Collaborator context in IScannerCheck

› You can’t generate your own Response from Collaborator server
Summary

2 Burp Suite Plugins
  › Burp-molly-scanner
  › Burp-molly-pack

Opensource
  › github.com/yandex/
Questions?
Contacts

Abakumov Andrey
✉️ a-abakumov@yandex-team.ru
🐦 andrewaeva

Zaitov Eldar
✉️ ezaitov@yandex-team.ru
🐦 kyprizel