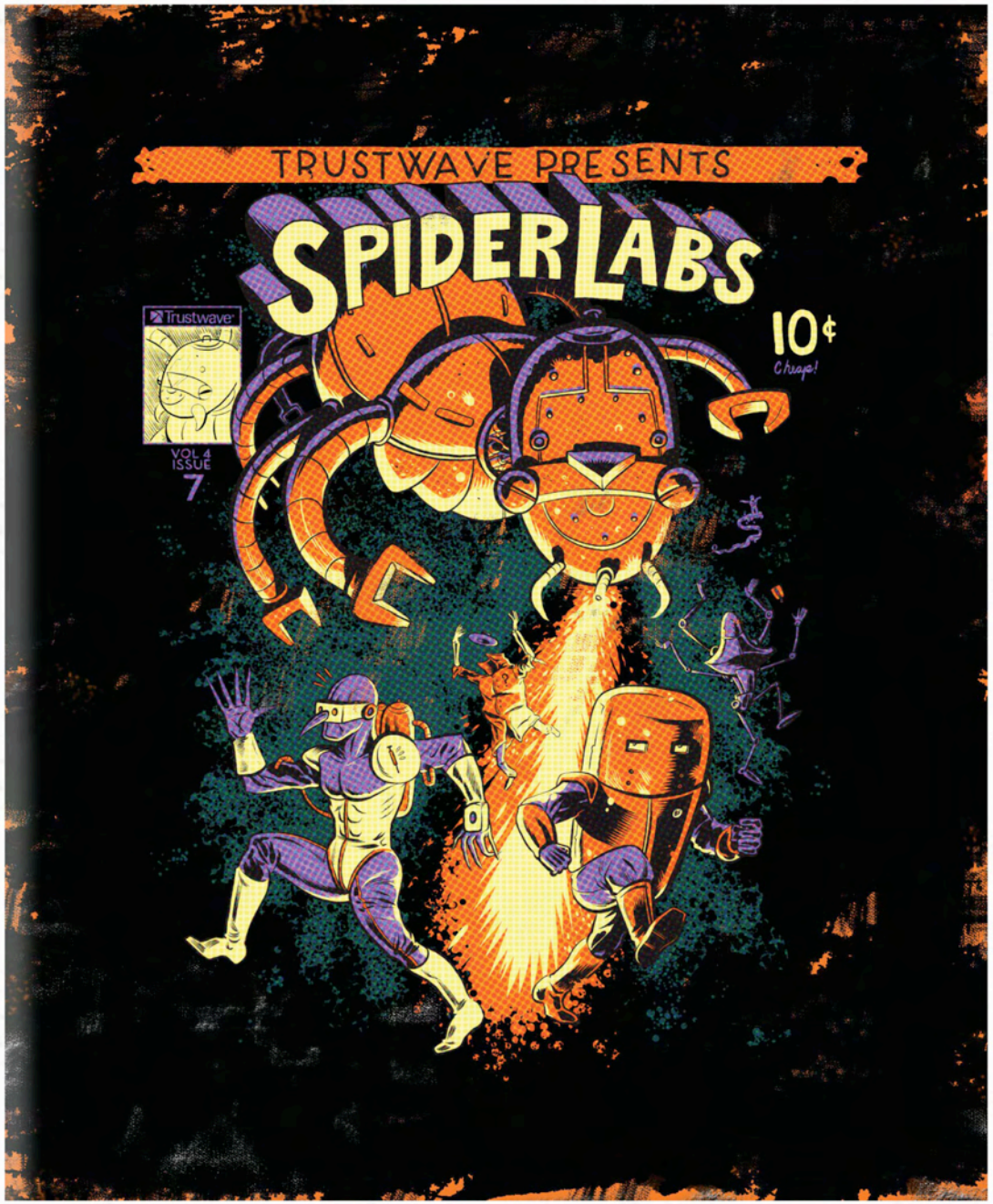


deblaze

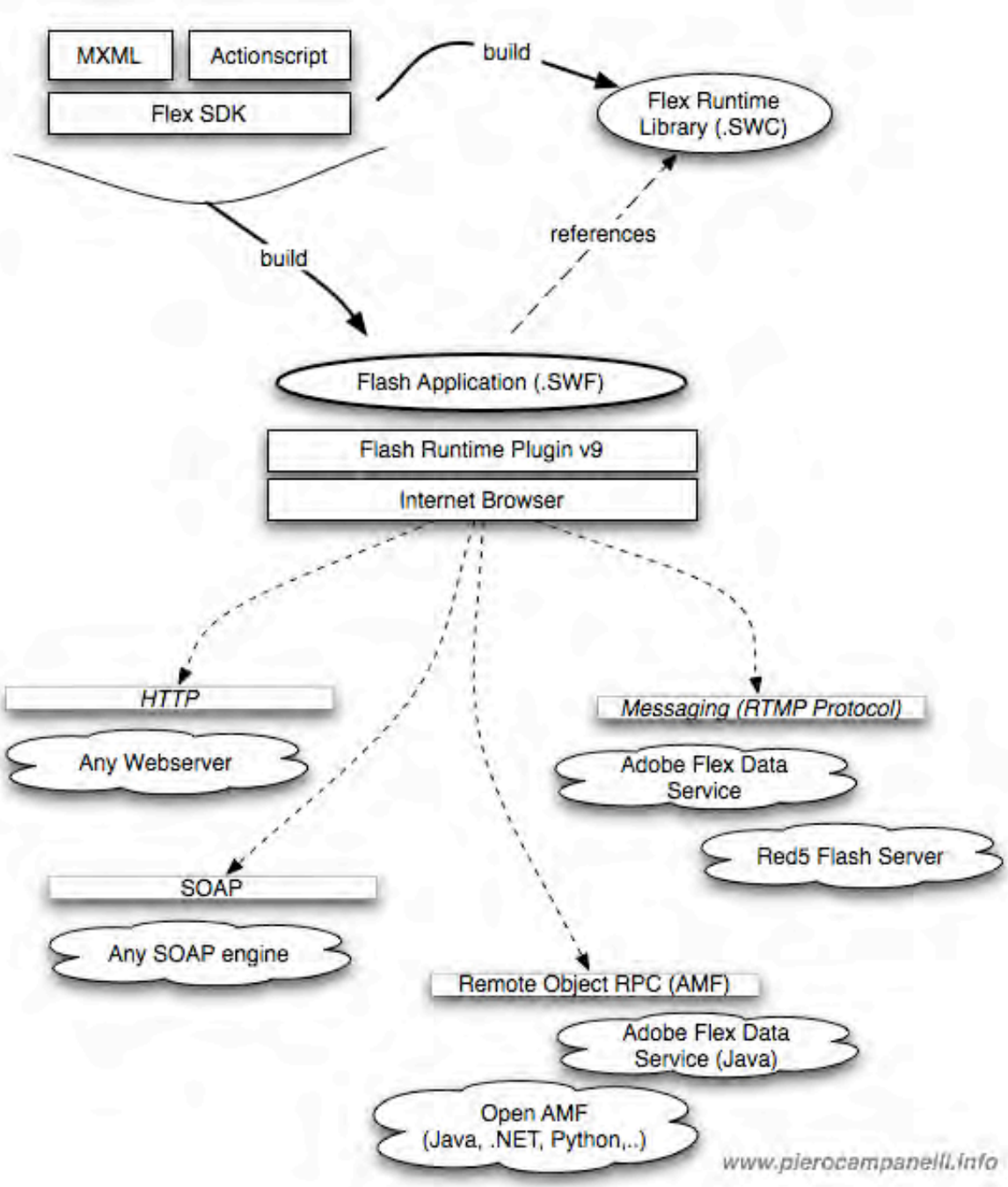
Trustwave®

Jon Rose
Trustwave's SpiderLabs

Flash Remoting Technologies

Flex Data Services

- Data Management
  - Update client and/or server when data changes
- Messaging
  - Real Time Messaging protocol (RTMP)
  - Pub-sub model
  - Real-time data streaming
- Remoting
  - HTTP, SOAP, AMF
  - Automatic data marshalling
- PDF
  - Create and edit PDF's


MXML
Actionscript
Flex SDK
build
Flex Runtime Library (.SWC)
build
references
Flash Application (.SWF)
Flash Runtime Plugin v9
Internet Browser
HTTP
Any Webserver
Messaging (RTMP Protocol)
Adobe Flex Data Service
Red5 Flash Server
SOAP
Any SOAP engine
Remote Object RPC (AMF)
Adobe Flex Data Service (Java)
Open AMF (Java, .NET, Python,..)
www.pierocampanelli.info

CrossDomain.xml

```
<cross-domain-policy
xsi:noNamespaceSchemaLocation="http://www.adobe.com
/xml/schemas/PolicyFile.xsd">
  <allow-access-from domain="twitter.com"/>
  <allow-access-from domain="api.twitter.com"/>
  <allow-access-from domain="search.twitter.com"/>
  <allow-access-from domain="static.twitter.com"/>
  <site-control permitted-cross-domain-policies="master-only"/>
  <allow-http-request-headers-from domain="*.twitter.com"
  headers="*" secure="true"/>
</cross-domain-policy>
```

Flash Remoting Insecurity

- Developers fail to restrict access to methods:
  - Authentication
  - Authorization
- Method & Service names can be brute-forced
- Flex servers can be fingerprinted
- Common vulns in remote methods:
  - Injections
  - Information leakage
  - Denial of service
  - Privilege escalation

Finding Methods & Services

- Decompile SWF and search for remoting calls
- Watch network traffic
- Dictionary attack against server

Finding SWF Remoting Calls

- Download SWF file
- Decompile
  - Sothink SWF decompiler or HP SWFScan
- Analyze ServerConfig.xml
- Regex for remoting methods

SWF Remoting ServerConfig.xml

- Often embedded in the SWF
- Provide URL's and service names
- Destination id represents services
  - securityService
  - exampleService
  - mathService

```
ServerConfig.xml = <services>
 <service id="remoting-service">
  <destination id="securityService">
   <channels>
   <channel ref="my-amf"/>
   </channels>
  </destination>
  <destination id="exampleService">
   <channels>
   <channel ref="my-amf"/>
   </channels>
  </destination>
  <destination id="mathService">
   <channels>
   <channel ref="my-amf"/>
   </channels>
  </destination>
 </service>
  <channels><br/>
   <channel ref="my-amf"/><br/>
   </channels><br/>
 </destination><br/>
 </service><br/>
```

SWF Remoting Search

- Search for remoting methods
  - send, service, remote, etc

findstr /I /N /S "sender\." *.as

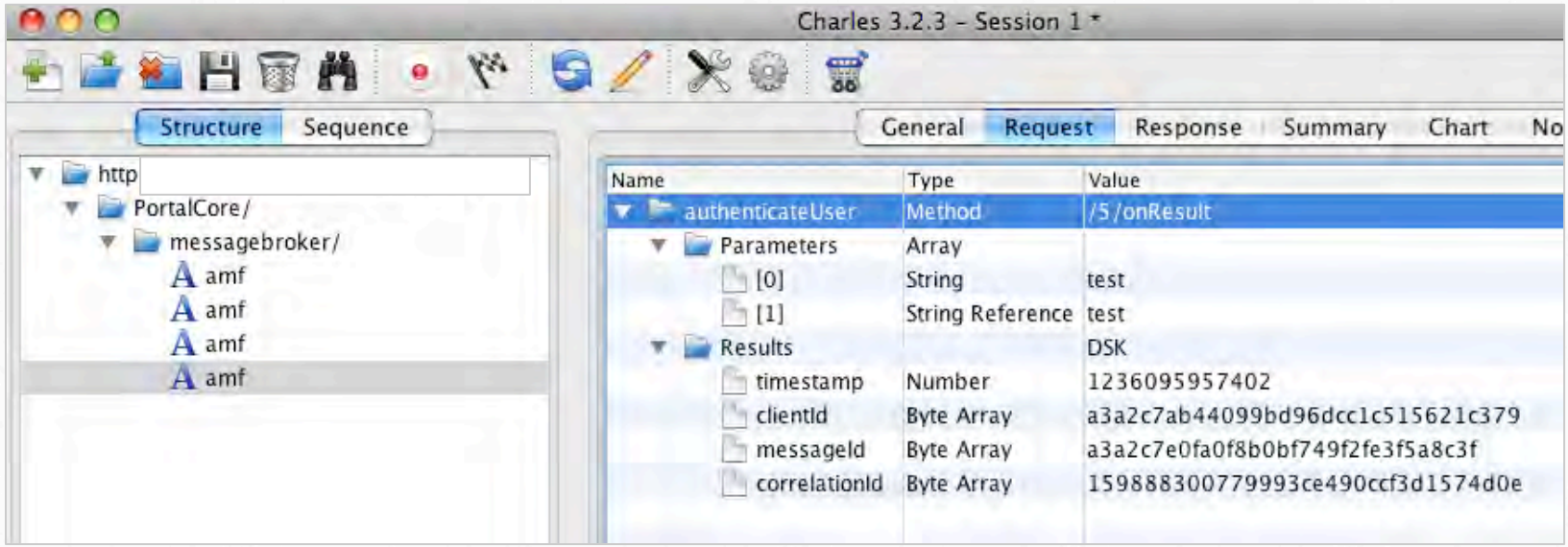
```
var _loc_6:* = _sender.startStreamingLog(param1, param2, param3, param4, param5);
_sender.startStreamingLog(param1, param2, param3, param4, param5).addResponder(_responder);
var _loc_6:* = _sender.createRequest(param1, param2, param3, param4, param5);
_sender.createRequest(param1, param2, param3, param4, param5).addResponder(_responder);
var _loc_6:* = _sender.installConfigVersion(param1, param2, param3, param4, param5);
_sender.installConfigVersion(param1, param2, param3, param4, param5).addResponder(_respond

var _loc_3:* = _sender.resolveRequest(param1, param2);
var _loc_2:* = _sender.getChanges(param1);
var _loc_5:* = _sender.joinDomain(param1, param2, param3, param4);
_sender.joinDomain(param1, param2, param3, param4).addResponder(_responder);
var _loc_4:* = _sender.commentOnRequest(param1, param2, param3);
_sender.commentOnRequest(param1, param2, param3).addResponder(_responder);
var _loc_1:* = _sender.getCategories();
```

AMF Network Traffic

- AMF is:
  - Serialized ActionScript object
  - Transported as HTTP POST body
- Charles proxy can intercept and decode AMF traffic
- Wireshark captures can disclose Url's, Services, and Methods
  - No native decoder

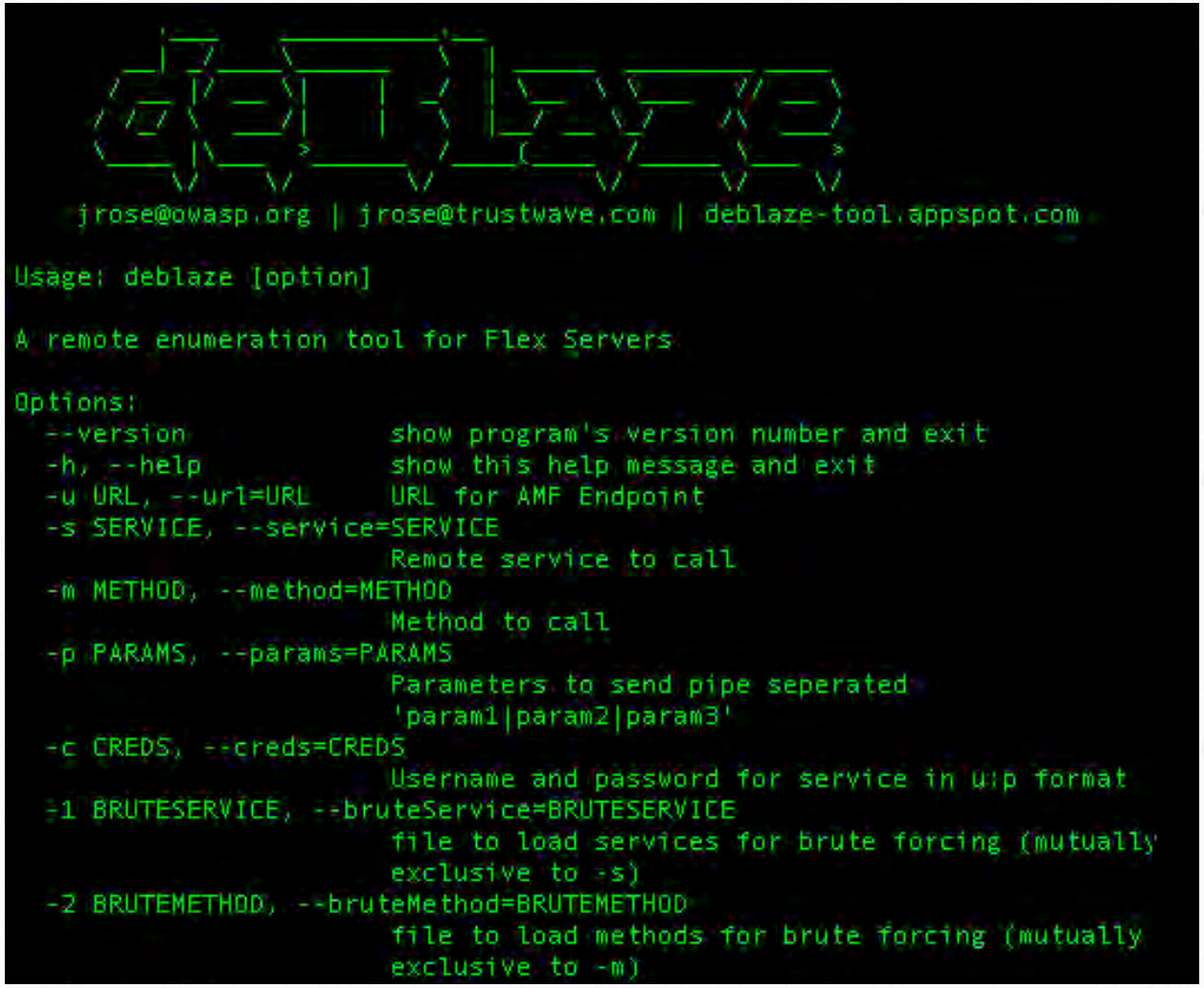
AMF Network – Charles Proxy



AMF Network – Wireshark

Dictionary Attacks

- Determine valid service and methods
  - Based on error messages
  - Fairly fast
  - Easily predictable method/service names
    - Login
    - getters
    - setters
  - Possible to build default wordlist


jrose@owasp.org | jrose@trustwave.com | deblaze-tool.appspot.com
Usage: deblaze [option]
A remote enumeration tool for Flex Servers
Options:
--version show program's version number and exit
-h, --help show this help message and exit
-u URL, --url=URL URL for AMF Endpoint
-s SERVICE, --service=SERVICE
Remote service to call
-m METHOD, --method=METHOD
Method to call
-p PARAMS, --params=PARAMS
Parameters to send pipe seperated
'param1|param2|param3'
-c CREDS, --creds=CREDS
Username and password for service in u:p format
-1 BRUTESERVICE, --bruteService=BRUTESERVICE
file to load services for brute forcing (mutually
exclusive to -s)
-2 BRUTEMETHOD, --bruteMethod=BRUTEMETHOD
file to load methods for brute forcing (mutually
exclusive to -m)

Securing Flash Remoting

- BlazeDS
  - Only public methods defined in remoting-config can be called
  - Use security-constraints in remoting-config.xml to each method
    - include-methods
    - exclude-methods
  - Read the Adobe BlazeDS security docs

Securing Flash Remoting

- AMFPHP
  - Methods that start with an underscore cannot be remotely called
  - Remove the Service Browser and DiscoveryService service
  - Disable remote tracing and debugging headers by setting PRODUCTION_SERVER
  - Use beforeFilter for authorization controls
- PYAMF
  - Enable authentication on the server

Questions

- Next Steps
- Future Research
- Latest Code
  - deblaze-tool.appspot.com
- Thanks
  - Spiderlabs
  - Nick Joyce
  - Stads9000
  - GDS crew

- Contact me:
  - jrose@owasp.org
  - jrose@trustwave.com