NATIONAL SECURITY AGENCY
CENTRAL SECURITY SERVICE

FORT GEORGE G. MEADE, MARYLAND

FOXACID SOP

FOR

OPERATIONAL MANAGEMENT

OF

FOXACID INFRASTRUCTURE

Prepared By:

Reviewed By:

Reviewed By:

Reviewed By:

TABLE OF CONTENTS

1.0 (S//SI) BACKGROUND/PURPOSE

Background.

Purpose. To publish procedures for checking XSS viability of all tasked domains used by the FOXACID team. This SOP should also contain checks for potential exposure of XSS in actions that can be taken after viewing.

2. (TS//SI) BILLET DESCRIPTION/ACE OBJECTIVES

Number	Objective
1	**Computer Network Exploitation Operations Support** **Description:** (TS//SI//REL) As a new member of the Production Operations Division's FOXACID mission, your primary objective is the following: - Learn the basics of the core FOXACID capabilities (Servers, Crafting, Deploying, Tags and Cross-Site Scripting) - Understand the purpose and functionality of your new TAO, Unix, FOXACID, WAITAUTO, RAISEBED, FORRESTPLACE, FOXHelp and PUZZLECUBE accesses. - Demonstrate an understanding of the TAO classification guide as it relates to the FOXACID mission. - Demonstrate an understanding of all key entities of TAO and how each corresponds to the FOXACID mission. - Receive FOXACID briefing from FOXACID team leader. - Demonstrate understanding of the FOXACID deployment categories and the various missions supported by the FOXACID team. - Demonstrate understanding between FOXACID deployment categories, special deployment process and deployment techniques. - Sit in at least one ROC stand up meeting. - Understand the roles and differences between FOXACID customer support and technical stewardship of TAO's primary initial CNE access capability. - Be assigned a FOXACID coach and mentor.
2	**CNE Operator, responsible for TAO exploitation infrastructure** **Description:** (U//FOUO) Responsible for the operational posture of the FOXACID infrastructure. Work with senior FOXACID team members to understand all aspects of the FOXACID servers, to include knowing the fundamental basics and operations specific to FOXACID. Fundamental basics include: - Server OS basics - System Administration Functions - XML. Specific FOXACID operations: - Missions supported by each FOXACID server - Filter management - Payload configurations - Tool testing procedures - Differences between exploit, wrappers and payload and the purpose for each one - FOXACID server timing integration - Data flow and server integration with CDR - Server checklist familiarization - Demonstrate understanding of FOXSEARCH applications - Understand the FOXACID scripts and how they pertain to the FOXACID servers - Immediate actions on servers for trouble shooting efforts.

3	**CNE Operator, responsible for unique mission support** **Description:** In addition to understanding how the FOXACID servers support traditional FOXACID missions, be able to identify the infrastructure, filters and processes to supporting the following: - YACHTSHOP (Targeted) - YACHTSHOP (Untargeted) - Man In The Middle (SECONDDATE, MAGIC SQUIRREL and MAGICBEAN) - FINKCOAT - FRUGALSHOT - Web Forum sessions - QUANTUM missions - FERRETCANNON (JSOC and CIA tools - i.e. BEACHHEAD) - OLYMPUS/UNITERAKE Team capabilities.
4	**CNE Operator, responsible for tool testing and implementation** **Description:** As a FOXACID team member responsible for the FOXACID servers, you are responsible for maintenance and functionality of the of FOXACID server Plugins, Exploits, Wrapper and Payloads. You are responsible for the following: Demonstrate an understanding of the exploitation process, capabilities and limitations and abilities to add, modify and update as it pertains to: - FOXACID Server Plugins - Exploits - Wrappers - Payloads
5	**Pursue Professional Development as an CNE operator** **Description:** As a CNE operator within TAO and the ROC, your knowledge will be tested and needed to troubleshoot systems, operations and potential exploitation errors. Being a constant student of your profession to improve the CNE Tactics, Techniques and Procedures (TTPs) currently used as well as growing you Knowledge, Skills and Abilities (KSAs) is a continuous process. You are highly encouraged to work with your Coach and Mentor to identify a professional plan for the next year that will include a mix of individual training and courses to increase your technical knowledge.

3. GENERAL FOXACID SERVER INFORMATION

3.1. (TS//SI) FOXACID SERVER BACKGROUND

The Foxacid server is a Microsoft Windows 2003 server that has the FA server software, FA plugins, and payloads installed. A series of filters, modrewrite, prefilter, and postfilter (in that order) are used to provide logic to the server. This logic determines whether a tag coming in might be changed, what payload a target might get, or a 404 or 200 if it is a blocked IP.

3.2. (TS//SI) FOXACID SERVER INTERACTION WITH CDR AND DATA FLOW

CDR is the method of data transfer used by the Foxacid servers. Data on the low side is transferred in an encrypted format to a receiving computer. The data is then transferred to the high side, decrypted, and placed in its appropriate file location. This includes the data that populates Foxsearch and our heartbeat files.

Sometimes CDR goes down and/or becomes backed up with an excess amount of data. There are many projects beyond Foxacid that use CDR and an error in any one of these projects can affect CDR. Depending on how long there is a problem with CDR will affect just how current the data is. The greater the backlog in data, the greater the amount of time it will take for everything to get back up to date.

3.3.(TS//SI) FOXACID SERVERS AND SUPPORTED MISSIONS

Server	***Mission***
XS10	YachtShop
XS11	GCHQ MITM
FOX00-6000	***Test Server (Spam)***
FOX00-6001	CT Spam
FOX00-6002	ME Spam
FOX00-6003	AA Spam
FOX00-6004	RU Spam
FOX00-6005	EU Spam
FOX00-6100	***Test Server (MITM)***
FOX00-6101	CT MITM
FOX00-6102	ME MITM
FOX00-6103	AA MITM
FOX00-6104	RU MITM

FOX00-6105	EU MITM
FOX00-6106	CT-MAC
FOX00-6300	***Test Server (Enchanted)***
FOX00-6401	CCNE China
FOX00-6402	CCNE Russia
FOX00-6403	CCNE Other

*******ENCHANTED Operations have been ceased as of week of 20100118*******

3.4.(TS//SI) Exploits

Exploitation is the process gaining control of a target's computer. The Foxacid server decides exploitation, but it can be influenced by the Foxacid Exploitation Tag and by the Modrewrite filter. Exploitation can fail at any point after the first contact. The first contact is a success in XSS. After that the process is in the arena of exploitation.

What is a success in exploitation?

- Payload delivered.
- 404 bad process or 404 already deployed or self deleted.
  - Not deploying a payload is a success in exploitation because to get to this point to make these determinations the server has to have a successful exploitation.
- 404 not vulnerable
  - Even though the server did not try to exploit the target this is still a success. The server stopped the process for a known reason, not the target for an unknown reason.

What is a failure?

- There is a failure in exploitation if there are contacts after the initial contact but there is not a well-defined end to the process by the Foxacid server. This means no Payload, 404, or 200 message.

What can cause exploitation failure?

- JavaScript turned off.
- Target has a slow connection.
- Target surfs off the page before the process can complete.
- Browser has extensions loaded. This causes certain exploits to fail.

3.5.(TS//SI) Wrappers

DireScallop- enables payload persistence against targets that are using the DeepFreeze products.

DeepFreeze is used in many Internet cafes to prevent changes to a computer's operating system and software applications. It is designed such that when a machine with DeepFreeze boots it always starts in the same state. DIRESCALLOP disables DeepFreeze without the need for a reboot. Implants can then immediately execute and re-enable DeepFreeze after gaining persistence. Automated deployment of DIRESCALLOP through FOXACID is supported.

DIRESCALLOP 2.0.0.2 emits executables. It does not emit DLL's. The payload passed to FINKDIFFERENT will be written to disk and then executed. That is why only -FD1 will work when wrapping it. Failures for -FD2 and -FD3 are expected and normal. To make DIRESCALLOP an executable, you have to edit the VAL payload file (config_dv.xml) that it's being thrown with. Open the file and change the wrapper name of "Disablevalor_SelfDelete_IncludePriv_CheckFirewalls.dll" to "Disablevalor_SelfDelete_IncludePriv_CheckFirewalls.exe" Make sure to save file and restart the service.

DisableValor- Exploitation can fail at any point after the first contact. The first contact is a success in XSS. After that the process is in the arena of exploitation.

3.6.(TS//SI) PAYLOADS

Payloads are the backdoors that are delivered when exploitation is successful. Requesting different payloads DOES NOT HELP EXPLOITATION.

Requesting a payload other than the default should be done with care and sound reasoning. Each change deviating from the default payload is a new line in a configuration file. Each new line in a configuration file is a new check for the Foxacid server to perform. Each check the server performs slows the exploitation process. Slowing down exploitation hurts you and other analysts.

Current Payloads

Validator 8.2.1.1 (default)

MistyVeal 11.0.4.3- This payload is only used for special request. It's used only when targets have authenticated proxies, PSP Evation, and or special coordination with Interactive Op.

MistyVeal is a larger implant than Validator. It has a configurable call back time that can be changed with a granularity of increments in Days, Hours, or Minutes.

MistyVeal cannot call out on the network on its own. It piggybacks on Internet Explorer to call out on the network. If Internet Explorer is configured to use a proxy, MistyVeal will be able to use the proxy. This is the major strength that MistyVeal has over Validator.

If a target is using FireFox and has MistyVeal on the box, MistyVeal does nothing.

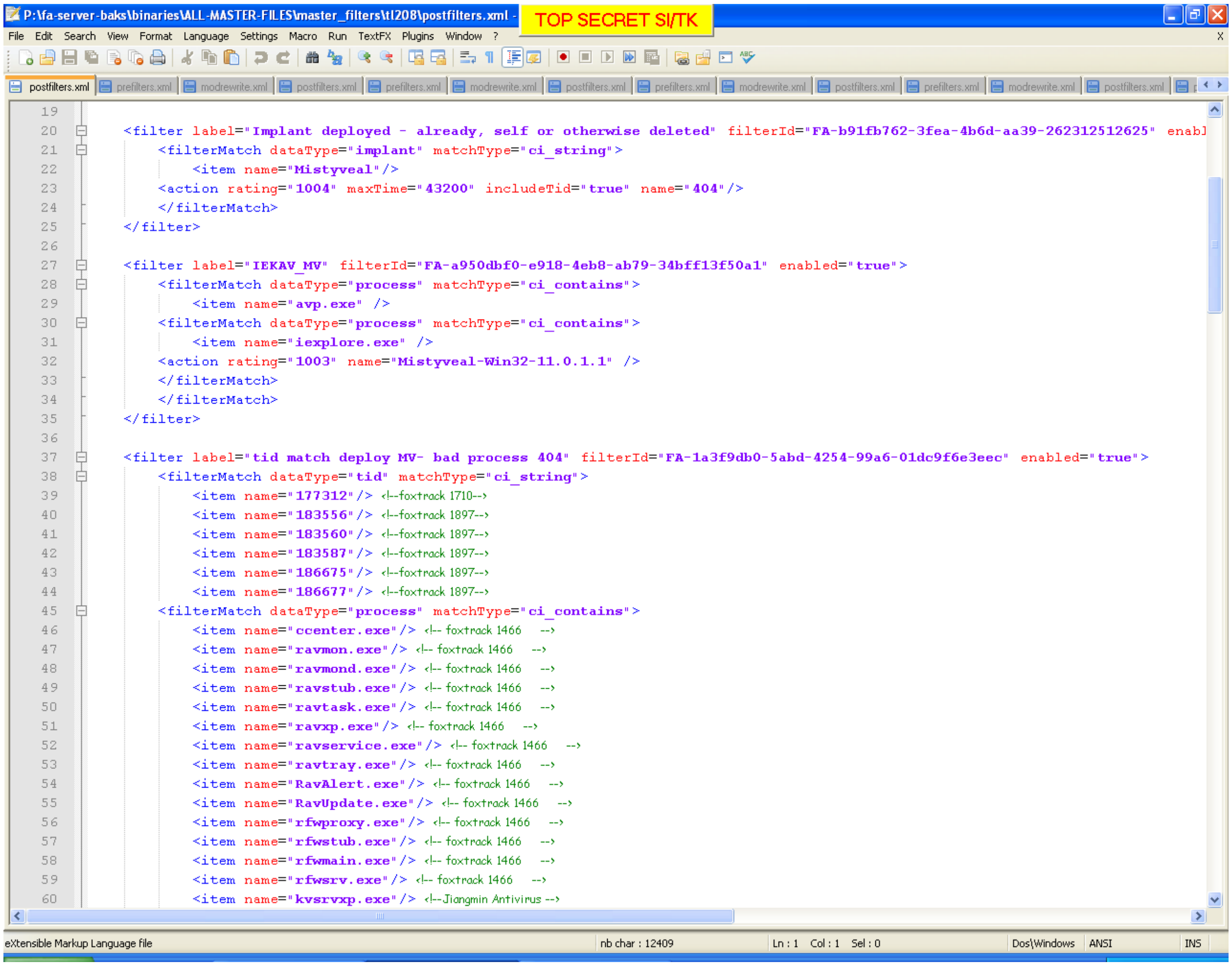
If a target is reading emails and Validator is not deploying, requesting MistyVeal for the target will not help. Requesting MistyVeal payloads DOES NOT HELP EXPLOITATION.

Special Configuration: For MV 11.0.4.3, change the config.xml file on the servers so that the overt site check reads http://search.yahoo.com instead of http://www.yahoo.com

Ferret Cannon- This is a "payload" that can throw any executable that is not normally thrown from a FOXACID Server, such as, United Rake, Peddle Cheap, PktWench, and Beach Head. Can throw both a .dll and an .exe. DLLs must use the –FD3 flag, but always test to be sure.

4. (TS//SI)

When a request is made to throw a payload other than the default, a change must be made in the filters on that specific server. If a filter type for that request is not present on the server, a filter must be created. All the filters are in XML format.



```
    <filter label="Implant deployed - already, self or otherwise deleted" filterId="FA-b91fb762-3fea-4b6d-aa39-262312512625" enabl
        <filterMatch dataType="implant" matchType="ci_string">
            <item name="Mistyveal"/>
        <action rating="1004" maxTime="43200" includeTid="true" name="404"/>
        </filterMatch>
    </filter>

    <filter label="IEKAV_MV" filterId="FA-a950dbf0-e918-4eb8-ab79-34bff13f50a1" enabled="true">
        <filterMatch dataType="process" matchType="ci_contains">
            <item name="avp.exe" />
        <filterMatch dataType="process" matchType="ci_contains">
            <item name="iexplore.exe" />
        <action rating="1003" name="Mistyveal-Win32-11.0.1.1" />
        </filterMatch>
        </filterMatch>
    </filter>

    <filter label="tid match deploy MV- bad process 404" filterId="FA-1a3f9db0-5abd-4254-99a6-01dc9f6e3eec" enabled="true">
        <filterMatch dataType="tid" matchType="ci_string">
            <item name="177312"/> <!--foxtrack 1710-->
            <item name="183556"/> <!--foxtrack 1897-->
            <item name="183560"/> <!--foxtrack 1897-->
            <item name="183587"/> <!--foxtrack 1897-->
            <item name="186675"/> <!--foxtrack 1897-->
            <item name="186677"/> <!--foxtrack 1897-->
        <filterMatch dataType="process" matchType="ci_contains">
            <item name="ccenter.exe"/> <!-- foxtrack 1466   -->
            <item name="ravmon.exe"/> <!-- foxtrack 1466   -->
            <item name="ravmond.exe"/> <!-- foxtrack 1466   -->
            <item name="ravstub.exe"/> <!-- foxtrack 1466   -->
            <item name="ravtask.exe"/> <!-- foxtrack 1466   -->
            <item name="ravxp.exe"/> <!-- foxtrack 1466   -->
            <item name="ravservice.exe"/> <!-- foxtrack 1466   -->
            <item name="ravtray.exe"/> <!-- foxtrack 1466   -->
            <item name="RavAlert.exe"/> <!-- foxtrack 1466   -->
            <item name="RavUpdate.exe"/> <!-- foxtrack 1466   -->
            <item name="rfwproxy.exe"/> <!-- foxtrack 1466   -->
            <item name="rfwstub.exe"/> <!-- foxtrack 1466   -->
            <item name="rfwmain.exe"/> <!-- foxtrack 1466   -->
            <item name="rfwsrv.exe"/> <!-- foxtrack 1466   -->
            <item name="kvsrvxp.exe"/> <!--Jiangmin Antivirus -->
```

When a new exploit or an update of an old exploit comes out, the newest versions of the plugins must be loaded on to the servers. Since the different plugin versions just add on top of each other, all the previous ones must be installed on the server. All software that needs to be installed should be done by MIT. Submit a RocHelp ticket if software needs to be loaded.

Payloads are updated as they are released from DNT and tested. We simply create a test tag using that payload and try to exploit one of our test images on our FORESTPLACE node. Occasionally certain payloads are considered too "dangerous" to be used globally so they become used for CT targets only.

We do not build a server from scratch, we receive a bare bones server that will run and MIT loads the FA software for us. This includes our server software, payloads, and plugins. MIT also uploads the VAL IDs, and installs the CDR keys. The FA team is responsible for updating the Payload ID file, uploading the new filters and configuring the payload config.xml files.

Once a month the server's times must be checked to make sure that they are all set to GMT +0, this ensures that when comparing logs, we don't have to keep on converting times.

8.0 (TS//SI) SERVER IMMEDIATE ACTIONS

When a server goes down there are many possible issues and many possible solutions. The easiest and most common issue is a CDR back up. Initially you will see one or two servers go down. Soon after you will see a few more go down or all depending on how backed up CDR is. Occasionally, CDR will fix itself pretty quickly and other times will stay down for quite some time. Normally if CDR is down for more than half an hour, a ROC help ticket would be put in.

If a server randomly goes down, an easy fix is to attempt to go into the services and restart the "FA: Server" service. In normal circumstances, if a server goes down it is usually an easy fix. One simple issue is that a heartbeat file could have been missed, and the server will look like it has gone down until the next heartbeat makes it through. If they are not making it through, make sure that the heartbeat command is still running and that the "tobesent" logs are making it through. If there is low disk space the logs will stack up and stop being pushed through and the server will go down.

If a server went down right after a filter update, best guess would be that there is an issue with that XML file. Most commonly there is a missing closing tag and if you try to open the file using Firefox, it will normally tell you where the issue lies. Other than that you must look through everything that you just updated to find where the issue might be.

9.0 (TS//SI) UNDERSTANDING FOXACID SERVER INPUT TO FOXSEARCH

FOXSEARCH is located at
https://[redacted]

A quick way to get information back is to enter the TLN/HMAC and Run Query.

Make sure the below parameters are also checked for FOX Contact Fields (See Figure below)

- User-Agent- specifies the OS and browser being used

- Payload ID- specifies the ID given to the payload that was dropped
- Proxy Type- specifies name of proxy

Possible Codes/Errors

- *Not Vulnerable-* This is an error that you get when the plugin was unable to exploit, and its not that it failed either. When the survey runs, it sees that the target browser may not have enough enable (such as flash). The exploit is not able to use these and is unable to elevate its own privileges, so the survey comes back as "Not Vulnerable". All the exploits use the same method for privilege escalation as well, so it does not matter which one you use.

- *Invisible Touch-* This is not an error. "Invisible touch" is a contact only from an inert image tag that we put into our spam messages. The purpose of this tag is to show that the spam message was opened by the target even though we may not have exploited the target's machine (i.e. the FOXACID tag possibly was stripped in delivery, etc.)

- *Implant deployed already or self deleted-* This is an interesting error, because it is not always as bad as it looks. When a box calls back to our servers, it goes through our filters to search for bad (or good depending how you look at it) PSPs and to see if there is already a payload present. When a payload self deletes for ANY REASON, it leaves an errata code, telling the next payload why it self deleted. These reasons could be a PSP that would catch it, self deleting due to not being able to call out in a timely manner, and a payload is already there. Now for the case of a payload already being there, it could be there and not calling out, or it is calling out however Foxsearch decided to put the contact info in a strange place. How Foxsearch places that information is in the date range of when the VAL/MV FIRST called out, not the latest callback. So this means that it could be almost all the way at the top of the page. A good way to look for this is watching for the text in the white boxes saying that there are x number of results but only two are displayed.

- *Group Not Found-* Most commonly there would be an issue with the tag used. Either it was wrong to begin with, or (most often) for some reason the group portion of the tag was changed from the point of creation to the point of actually throwing it. Also, occasionally with the EVAK plug-in, the error will show despite the fact that the plug-in was able to build and a payload was even dropped. In this case the error is just a fluke.

10.0 (TS//SI) LOADING SERVER SOFTWARE, FILTERS AND TESTING

Loading FOXACID Server Step by Step

Once the new server is received, follow the below process to have MIT build the FOXACID Server.

- Request CDR Keys and domains for new server
- Create filters and directory in Clearcase for the new server
- Update the fa_install (located in Clearcase) with new server information
- Submit a RocHelp ticket for MIT to install Software in the following order: Server ***Plugins***Payloads***Val ID
- Install "keys" into C:\main\keys\
- Edit C:\main\config\server.xml
  - Updates include: CDR info (Verify IP and port, TA_ID, TE_ID, and IN_ID), list begin, list ends, set log forwarding to false
- Verify C:\main\config\deployment_types.xml
- Edit each payload config file
- Update C:\main\payloads\config\ids\[PAYLOAD_ID].txt (Ex. Server FOX00-6001 payload id – 600100000:600199999
- Run fa_build_ops_disk.pl from Clearcase to upload new filters to thumbdrive. Then load updates on RAISEBED/WAITAUTO. If you're logging on to RAISEBED, upload from command prompt by running fa-install.pl and select necessary server. If you're logging into WAITAUTO, you need to log in to the specific server that you're trying to update and then upload from the command prompt.
- Have MIT schedule "server heartbeat".
- Edit C:\main\config\server.xml
  - Start log forwarding
- Ensure logs are being moved through CDR

Testing FOXACID Server build

- Log on to a virtual machine
- Open a command prompt to run the below command from the desktop location
  - info –prep (cleans the box)
  - info –check (shows if the box is clean of exploits)
- Open a browser on a virtual machine.
- Open task manager to observe the current processes running
- Test payload with a test tag and use a TLN from the 99xx series and msgid z1zzz
- After execution is complete, clean the box once more and re-check it.

Creating/Modifying Filters in Clearcase

In order to create filters in Clearcase, you will need to submit an ITSC ticket for Rational Clearcase Explorer 7.0.0.1.

Once installed and configured, expand your view (your sid) within Clearcase and drill down to Ops_Mgmt/FOXACID/Server/binaries/ALL-MASTER-FILES/master_filters. You will see all of the servers and within each folder there will be the mod, pre, and post filters.

To make changes to the filters, you have to select the server you want and then select the filters by right clicking them to select "Check Out". Enter a brief message to say why you're checking it out and click "Ok." To know you checked out a filter, there will be a green check on the filter to let you know that it has been checked out. Right click the filters again to open them in Notepad++.

In order to understand how to create filters, you need to understand the order of the filters and what they each are used for. The modrewrite is where we can re-write the tag to make it look like something else or hit on a different group. Next, you have the prefilter, which is used to whitelist and blacklist any IPs/HMACs/TLNs that you want to proceed thru the prefilter or stop at the prefilter. Last, the postfilter is where exploitation occurs, so this is where all payloads are loaded.

In every filter you will have the following pieces:

- *Filter Label*- this is a label of your choice that tells you what the filter is for/doing
- *Filter ID*- this is the unique ID (guid) that can only be used once on a filter. The guids are found in the binaries folder in the lotsofguids---USE_THIS_FILE_FOR_NEW_Filters.txt file
- *Enabled*- this is used to set the filter to "true" or "false". True makes the filter active on the server and false makes it inactive
- *FilterMatch datatype*- this tells you what type of information you will be filtering on
- *MatchType*- this will be ci_string or ci_contains
- *Item Name*-This item will be the information that you are trying to capture, such as IP, TLN, MAC, path, and/or process
- *Action Rating*- the order in which the filter will be read. It hits the filters from highest to lowest
- *Name*- this determines how the filter will continue
- *Flags (Optional)* An optional attribute that is commonly used for payload wrappers like DIRESCALLOP and DISABLEVALOR. The latest release of FOXACID also uses this attribute to flexibly change the payload instantiation method. This is extremely useful for PSP evasion.

  -FD1 : Execute EXE (current default behavior)

-FD2 : Load DLL directly into memory
-FD3 : Inject DLL into svchost.exe (should be set to this)

- *Thresholding (Optional)* An optional attribute that can be applied to a specific filter or to the entire set of filters. Setting a default threshold is done at the top level of the XML as an attribute to filters. Otherwise it is included as an attribute to the action element. Thresholds can be set per initial contact (maxThreshold), or specific to an amount of time. If paired with a deploymentType filter (for example), the threshold can also be checked against the Target and/or Deployment ID. Once a maxThreshold has been met, all contacts will receive a 404 until the threshold is cleared.

  Example: `action rating="1000" maxThreshold="10" maxTime="5" includeTid="true" name="go"`

Every Filter Match needs to have a closing Filter Match.

Below are example filters from each filter type.

MODREWRITE

```
<filter label="PF Mod" filterId="FA-9159b439-206d-4529-98ac-8a3076142688" enabled="true">
   <filterMatch dataType="get" matchType="ci_string">
     <item name="/y/rotate/img128484 html"/>
     <action rating="927" name="/demo/lib/python/1/queued/display9952_z2zzz.html"/>
   </filterMatch>
  </filter>

  <filter label="PF Mod Touch" filterId="FA-4c242e0b-dcc3-4bd3-b4ec-e15fe0db12c1" enabled="true">
   <filterMatch dataType="get" matchType="ci_string">
     <item name="/y/rotate/img89832.jpg"/>
     <action rating="926" name="/demo/public/stuff/pen/tar/full9952_z2zzz.jpg"/>
   </filterMatch>
  </filter>
```

PREFILTER

```
<filter label="hb_tid_match" filterId="FA-1599b468-145a-45f2-bde1-7e8cec128052" enabled="true">
            <filterMatch dataType="tid" matchType="ci_string">
                        <item name="9911"/>
            <action rating="1000003" name="go"/>
            </filterMatch>
</filter>

<filter label="Invisible Touch" filterId="FA-7dceaf90-4250-42b0-b87f-c6e67aab5602" enabled="true">
       <filterMatch dataType="gid" matchType="ci_string">
                        <item name="pen" /> <!--  pen should point to default0.xml -->
       <action rating="999994" name="200" />
       </filterMatch>
 </filter>
```

```
<filter label="404 TLN" filterId="FA-7dceaf90-4250-42b0-b87f-c6e67aab1111" enabled="true">
    <filterMatch dataType="tid" matchType="ci_string">
            <item name="9934" />
    <action rating="9995" name="404" />
    </filterMatch>
 </filter>
```

```
<filter label="TID DID go" filterId="FA-ea885340-fa96-4a20-8fe0-99466baa0dcb" allContacts="true"
enabled="false">
            <filterMatch dataType="did" matchType="ci_string">
                    <item name="z1zzz" />
            <filterMatch dataType="tid" matchType="ci_string">
                    <item name="9952"/>
                    <action rating="8999" name="go"/>
            </filterMatch>
            </filterMatch>
        </filter>
```

POSTFILTER

```
<filter label="WAK TID-deploy Val" filterId="FA-06198d50-f66e-4bd5-969d-0dcf87b95a41" enabled="false">
        <filterMatch dataType="tid" matchType="ci_string">
                <item name="9952"/>
                <action rating="1029" name="Validator-Win32-8.2.1.1"/>
        </filterMatch>
</filter>

<filter label="DMWH Survey" filterId="FA-d612efea-c1b1-4634-a209-5fea32c6bdd9" enabled="true">
        <filterMatch dataType="callpl" matchType="ci_string">
                <item name="DMWI"/>
                <action rating="1028" name="404"/>
        </filterMatch>
</filter>

<filter label="DF deploy DSVAL" filterId="FA-48bc3217-ce89-4762-80c8-341a36b2bc1e" enabled="false">
        <filterMatch dataType="process" matchType="ci_contains">
                        <item name="FrzState.exe"/>
                        <item name="DFServEx.exe"/>
                        <item name="FrzState2k.exe"/>
                        <item name="DF5Serv.exe"/>
                        <item name="FRZSTATE9X.EXE"/>
                        <item name="FrzState.exe"/>
        <action rating="1018" name="Validator-Win32-8.2.1.1" flags="-DSDirescallop-Win32-2.0.0.2 -FD1" />
        </filterMatch>
</filter>
```

11. (TS//SI) LOADING FOXACID PLUGINS AND TESTING

Loading FOXACID Plug-in Step

- Submit a RocHelp ticket to get a new plugin version installed on FA Server.

Testing Plug-in

- Read over user docs:
  - Identify the purpose of the implant
  - Identify what OS/Browser the implant can be thrown against
  - Identify LP version implant works with
- Create filter to deploy implant using test TLN
- Create test tag
- Deploy implant on test box (Forest Place)
- Verify implant was successfully deployed
  - Check for implants process name in the process list
  - Run info –check on the command line
  - If implant did not deploy successfully check FOXCONTACT logs for errors and redeploy
- Identify payload ID and VAL LP the implant will call back to. Take this information to the OLY team and verify call back
- Task OLY/UR team to remove implant from test box.

*****Please make sure that the GEN Plugin is loaded on ALL spam servers. This can be retrieved from the c:\\temp\(most current plugin backup date)\config\gen.xml This file allows for the spam messages to be sent successfully. They may be GEN plugin files on certain MiTM servers, but not all.*****

12.0 (TS//SI) LOADING FOXACID PAYLOADS AND TESTING

Loading FOXACID Payload Step by Step

- Submit a RocHelp ticket to install new payload on an FA Server.
- Once loaded, configure the config.xml file with LP and callback times

Testing FOXACID Payloads

- Create new filter to include 99xx for testing
- Upload new filter to WAITAUTO or RAISEBED using Clearcase
- Log on to a virtual machine
- Open a command prompt to run the below command from the desktop location
  - info –prep (cleans the box)
  - info –check (shows if the box is clean of exploits)
- Open a browser on a virtual machine.

- Open task manager to observe the current processes running
- Insert a test tag with test TLN for the desired server and run it
- Observe the task manager to see if the new plug in runs
- After execution is complete, clean the box once more and re-check it.

Payload TEST TLNs

9920 – 9959	Test Spam

Test TLNs can be found at https:/[redacted]

13.0 (TS//SI) FOXACID SERVER JQR

FOXACID Server JQR is located on foxacid NFS9 under FOXACID SOPS\JQRs

14. (TS//SI) SPECIAL CASES

Loading SSL certificates on FA Server

Contact MIT to inform them that you will be loading SSL certificates on servers.

- Insert CD that has SSL certificates loaded on them into PC.
- Log on to server that you will be uploading the SSL certificates.
- Copy all certificates from CD into the C:\main\keys directory
- There should be two important files: cert and key
  - Open each file in notepad to see which is which, the key file will begin with "BEGIN RSA PRIVATE KEY" and end with "END RSA PRIVATE KEY". If any other information is before or after that, you can delete the information. The cert file begins with "BEGIN CERTIFICATE" and ends with "END CERTIFICATE". If any other information is before or after that, you can delete the information.
  - Rename the cert and key files. Whatever names are already assigned add _key.pem to the key file and _cert.pem to the key and certificate files respectively. Here is an example: Original file names- whatever.com.privkey and whatever.com_cert key; New names- whatever.com_key.pem and whatever.com_cert.pem
- Once certificates have been renamed, edit the server.xml file located at C:\main\config
  - Set SSL_ENABLED VALUE= true

  - Set SSL_IP VALUE with the external IP of the server you are adding the certificates to
  - Make sure the SSL_PORT NUM_VALUE = 443
  - Copy the exact names of the key file and cert file into the SSL_KEY and SSL_CERT VALUES. These names will come AFTER the backward slash.
    - SSL_KEY VALUE= “keys\whatever.com_key.pem”/>
    - SSL_CERT VALUE= “keys\whatever.com_cert.pem”/>
- RESTART the FAServer service

Decrypting files for Beachhead

This is the proper way to decrypt files for BEACHHEAD. When you log on to a Linux terminal, open a shell. No Quotes are used in the command shell.

- Mount your thumbdrive by entering “mz” then press ENTER and then type “cd /mnt/zip” then press ENTER again.
- You can view the directory that you are in by entering “ls” to list
- If your files are located in another folder on the thumbdrive, you need to cd into that directory. For example, if you have a folder called BEACHHEAD, you can get to the directory by “cd BEACHHEAD” press ENTER and then view the contents to make sure you are in the right folder.
- Once you are in the proper directory, make sure all of the files you need are in the same place. You will need the decrypt file “gdi32.exe.gpg” and all the files that need to be decrypted.
- If the file that needs to be decrypted is called “wtime_PIG.gpg” and you want to make it a .dll, enter the below command
  - Enter command “gpg -o wtime_PIG.dll –d witme_PIG.gpg” press Enter
  - Enter the passcode provided
- After this command is entered, the .dll file will be added to the directory that you are working from
- Before removing your thumbdrive, you have to unmount it.
  - “cd” then press ENTER (no quotes)
  - “uz” then press ENTER (no quotes)

Unzipping files for Beachhead

- Mount your thumbdrive by entering “mz” then press ENTER and then type “cd /mnt/zip” then press ENTER again.
- You can view the directory that you are in by entering “ls” to list
- If your files are located in another folder on the thumbdrive, you need to cd into that directory. For example, if you have a folder called BEACHHEAD, you can get to the directory by “cd BEACHHEAD” press ENTER and then view the contents to make sure you are in the right folder.

- Once you are in the proper directory, make sure all of the files you need are in the same place.
- You want to enter the below command then press enter. The files will extract to the directory that you are currently in
  - tar xzrf newlanders.tgz

Queuing VAL for Beachhead

When a VAL ID is queued through OLY/UR, make sure they have "execute after put" and "stop processing" selected.

To verify the test is calling back to the appropriate IP, run netstat –ano 5 inside a command prompt.

- If for some reason you are having trouble queueing a file from OLY, try to see if you can run the .dll manually from the Forestplace VM for testing. To run the file manually, enter the below command in a command prompt
  - Rundll32.exe val.dll, Init test.txt (*val.dll is the file you want to run and test.txt is whatever name you want to name it*)

POCs for Beachhead

[redacted]

Yachtshop Tasking/Detasking

The Yachtshop tasking tool can be accessed via the Puzzlecube home page. It is located under the TAO Tasking tools. When new tasking (via email) comes in for Yachtshop, the Yachtshop tasking tool is used to build the PENDING TASKING. Once the pending tasking is executed, clearcase needs to be updated.

Open a command prompt and build tasking for YS. *Example command below.*

```
M:\qjjenki\Ops_Mgmt\FOXACID\Server\Scripts\fa_build_ops_disk.pl –buildys
```

Select the most current date for Yachtshop that was executed.

After tasking is finished being built, load it on to server XS10.

- Log on to Raisebed and open a command prompt.

- Map to your thumbdrive.
- Run the fa_install.pl –r
- Enter the password for server XS10.
- Filters will begin to copy and service will automatically be restarted.

ISP MAC Assignment for Yachtshop

- Go to the Yachtshop Tasking tool to assign a new ISP MAC.
- Select the button at the top of the page that says "To ISP MAC Assignments Page"
- Which ever assignment is highlighted in red, from the drop down menu for New Assignment, select server XS10, and at the bottom of the page click the button that says, "Make New Assignment"

CI_Contains vs CI_String

These two are similar, but will pull back different information.

CI_Contains is where you are grabbing a portion of the tag to match on.

```
<filter label="MAX MOD" filterId="FA-73e8a18f-dfbc-49b2-8cdc-f33823866d9c"
enabled="true">
        <filterMatch dataType="get" matchType="ci_contains">
                <item name="/structs/segsm/bins/1/define/"/>
        <action rating="999" flags="merge" name="/app/views/bdb/1/calls/"/>
        </filterMatch>
</filter>
```

- The above filter shows how the tag will come in as http://domain/structs/segsm/bins/1/define/deploymentid[TLN]_msgid.html but will be rewritten to http://domain/ app/views/bdb/1/calls/deploymentid[TLN}_msgid.html

- In this case, the tag just needs to contain /structs/segsm/bins/1/define/ in order to be rewritten.

CI_String is where you are grabbing the exact portion of the tag.

```
<filter label="fizzle_1268257984 Wed Mar 10 17:39:57 2010" filterId="FA-40a32d4c-
ee69-4830-b7e7-046236a3ac2c" enabled="true">
        <filterMatch dataType="get" matchType="ci_string">
                <item name="/rMqhMYjZMaxsnleaIxdyZVFjjbAnuUzUKUXNnO.xml"/>
        <action rating="100" name="/bases/loaded/callers/1/ctrl/japp9955_z1zzz.xml"/>
        </filterMatch>
</filter>
```

- The above filter shows how the tag will come in as *http://domain/ rMqhMYjZMaxsnleaIxdyZVFjjbAnuUzUKUXNnO.xml* but will be rewritten as http:// domain/ bases/loaded/callers/1/ctrl/japp9955_z1zzz.xml

- In this case, the tag just needs to have the EXACT url in order to be modrewritten

<u>Loading Modrewrites</u>

1. Check out the modrewrite(s) that you will be updating from Clearcase. *(You may have to check out the prefilter and add the did if this is not a test.)*
2. Make sure thumb drive is plugged in.
3. One of the crafters will place batch scripts in the folder appropriate for the server required *(i.e. Huffmush2-3.6002.me for FOX00-6002)*.
4. Move each batch script labeled "modrw.on" to the "In Progress" folder.
5. Double click on each batch script.
6. Move each batch script to the "Finished" folder.
7. Double click on the "moveme.bat" script in the "In Progress" folder.
8. Fill out the information asked by the script *(press enter for any troubleshooting step).*
9. Check the modrewrite.xml file back into clearcase for the server. *(Check in the prefilter if necessary)*
10. Use the fa_build_ops_disk.pl script in the "Scripts" folder in clearcase to update your thumbdrive with the latest data.
11. Plug flash drive into the WAITAUTO server box.
12. Remote Desktop into the server(s) that need to be updated.
13. Use the fa_install.pl script on the flash drive, typing in all information required (i.e. F:\fa_install.pl -r), press ENTER, select server, and then select update information.

14. Double check the service is started if not, restart the FA service.

Frugalshot/DementiaWheel Tags

(TS//SI) FrugalShot is a TAO infrastructure component that supports exploit callbacks. The FrugalShot server is actually a FOXACID server running a set of plugins compatible with deployments of exploits for removable media, e.g. EasyHookup. In particular, the server uses FIDO, EPL and FRDO plugins. The use of the FIDO and EPL plugins provides the capability of delivering a payload upon receipt of the initial exploit callback. The use of the FRDO plugin provides the capability to capture DementiaWheel data exfiltrated by subsequent exploit callbacks.

(TS//SI) There can be multiple FrugalShot servers. Currently, there are two main operational servers:

- *FS501*: This server supports callbacks from the CASTLECREEK deployment. The EPL plugin must be for v1.5. In addition, the version of FinkDifferent embedded within the payload does NOT support the downloading of a DLL, i.e. it only supports the downloading and execution of an EXE payload. There are particular cleaner payload files created specifically for use with servers supporting CASTLECREEK data.

- *FS502*: This server supports callbacks from non-CASTLECREEK deployments. It too requires an EPL plugin supporting v1.5. However, the embedded payloads used for these deployments use a slightly newer version of FinkDifferent than was used for CASTLECREEK and can support the downloading and execution of EXE or DLL payloads. There are particular cleaner payload files created specifically for use with servers supporting non-CASTLECREEK data.

FS501

IPs are added to the server to receive DementiaWheel callbacks. No tags are generated for this.

Tags for FS502

Tags for FS502 are generated by using the below format. The parts highlighted in blue should be switched out each time for new tags.

http://hostA.xxuz.com/step/QueryRecord9952_z1zzz.html (survey)

http://hostA.xxuz.com/gens/QueryRecord9952_z1zzz.eot

- step and gens- these are modrewrite mappings. Located on the modrewrite.
  - URLs must be mapped to begin with "/fink/disco" and have a ".html" suffix to be forwarded to the FIDO plugin.
  - URLs must be mapped to begin with "/data/xml" and have a ".eot" suffix to be forwarded to the FRDO plugin.
  - As with any FoxAcid server, the remapped URLs must use group delimiters chosen from the "LIST_BEGIN" and "LIST_END" values within the server.xml configuration files.

- QueryRecord- Deployment type; You can switch between QueryRecord and DisplayList

- 9952- TLN of the project

- z1zzz- Message ID associated with tag. You can grab a new message ID from the Deployment Categories excel spreadsheet.

Troubleshooting Willowvixen

Directory listing error

- Log on to willowvixen server and verify that the domain being requested is pointed to the appropriate virtual directory
  - Double click IIS, drill down to web sites, locate virtual directory that should reference the domain of interest
  - Right click the virtual directory, select properties, go to Web Sites tab and select advanced. Look to see if you see the domain of interest. If its not there, add it
    - IP address- "Enter WV IP"

    - TCP port- 80
    - Host header value- “Actual domain name”
  - Next verify that the below 4 items are enabled in the default content page. If the below items are missing add them in.
    - index.htm
    - Default.aspx
    - Default.htm
    - Default.asp

<u>Loading certificates on FA Server</u>

1. Request SSL certs and a domain by submitting a ROCHELP ticket. Should specify name of the domain and the IP of the server that you want the domains on.
2. Once the certs have been created, they will need to be obtained either by “go dropbox” thumbdrive, or what has normally done, CD. Obviously you will need to physically go to the person who created the certs if you are going to use a CD or a thumbdrive. This is done by someone within MIT.

15. (TS//SI) FOXACID Tools

<u>Project Tracker</u> https: [redacted]

1. The Project Tracker is our current management tool for all the different projects that Foxacid supports.
2. This is a simple web page made from JavaScript and some PHP. There is no back button here. The color background tells you what privileges you have. Red is read only, Green is modify, Black is admin. To escalate privileges contact one of the admins ([redacted])
3. In the Project Track you can:
    i. Manage Current Projects (server specific, when trying to view projects click on the “All Projects” button)

ii. Add a Filter
iii. Add a Project (This will be used most by normal operators)
iv. Add a server
v. Add IPs of servers
vi. View projects that will start/end soon (3 days)

4. When adding a new project to the Project Tracker use this format (PROJECTNAME_TLN).
5. Under each project there is information space, which is blank by default when a new project is created. Information must be entered manually using the buttons below the information table.
6. Each of the buttons do exactly as they say. Add/Delete HMAC, TLN, IP and so forth.
7. After clicking on a button to add/delete information from a project, you must click the "Go Back to Project" button to return to the default project page.
8. DO NOT USE THE BACK BUTTON!!! It won't work. This is all technically one page.

Tag Maker https: [redacted]

1. The Tag Maker is separate from the Project Tracker. Any servers/domains that were added to one must also be added to the other. Buttons on the left allow you to add tags, domains, and servers.
2. To add a tag click on the "Add a Tag" button.
3. Add in the Project Name (all caps), select the server, add a TLN or a place holder "[TLN]/[HMAC]" if there is no TLN (if the Op will be using HMACs), and MSGID.
4. For MSGID you can use either a normal MSGID from \\Nfs9\foxacid\docs\DeploymentCategories.xls
5. OR if the project is going to be using SECONDDATE, you must use the "ace02468bdf13579" MSGID. This is mandatory in all SECONDDATE operations. This creates a date time stamp when the tag is being used. This time stamp prevents constant re-exploitation from the target hitting the back button in their browser.
6. To reference other tags on the server, click "View Server Tags".
7. To reference all other tags, click "View All Tags"
8. When creating a tag, there are drop down menus to select each portion of the tag.
9. Domain: Completely arbitrary.
10. Path/Plugin-type: Also completely arbitrary
11. List Begin/End: Again, arbitrary. NOTE: When you select the List Begin, it will automatically select the proper List End.

12. Group: Only one group to select from. "1". If another group is needed it can be edited before sending the tag off to the analyst/operator. Done only in special cases since default for every server is "1" group.
13. Deployment Type: While selecting anyone of these will not render the tag useless for certain ops, the portion in ( ) is what should be selected for the appropriate type of operations. EX. A SECONDDATE Operation or MAGICBEAN should be using the "(WEB)" Deployment Types. YATCHSHOP tags will use "YS" Spam tags should have "SPAM" and QUANTUMINSERT should have "QI".
14. Once all the fields have been selected, click "Create Tag" to fully view the tag you just created. Make sure there are no spaces.
15. "Save Tag" will then save the tag to the system.
16. To Edit tags, either view tags or view all tags and next to each tag there is an EDIT button. Click the button and you can edit any portion of the tag. This will reset all fields to default, so be sure to re-select fields you want to stay the same. The tag you are editing will appear at the top of the page.

FROZENGAZE/FABULOUSFABLE (Also FABFAB)

1. FABULOUSFABLE is used in automated SECONDDATE tasking. Instead of an interactive Op to put up inject rules, WATCHER tips hit the FROZENGAZE system and FABFAB triggers on selectors (MD5 hashed user names). By default rules will go up for 15 minutes (900 seconds) and after that time, the rule will be terminated.
2. Login info: login using Putty or to easily transfer files WinSCP3 popo00-213.home.local (10.0.80.213) User name: root Pass [redacted]
3. Each project is found under /fg3.0/config/(HEX values). 0x92…. Is DARKFIRE 0xdd….. is a CCNE project. 0xf4f is CRYPTICSENTINEL. DARKHELMET has three 0x5a3cb41c/d/e
4. Under each project are the selectors that can be matched on. You can tell a selector is active by looking at the file and seeing three lines separated from the main file that say ACTION=FAB, RULES=1,2,3,4,5(or what ever rules they want put up), and TIMEOUT=900 (or however long the rules need to stay up).
5. Each of the selector files are the same with the exception of the name being specific to a user name (remember MD5 hashed) and the FAB rules at the bottom. So when creating new ones you can just cp one of the other files that has the info you need. The thost_ file is the default action for any selectors that hit that project but don't have a selector file.
6. Each selector has a start and a stop file, one more directory deep in /xml. These are all the same with the exception of the name being selector specific. You can cp the start_template.xml and stop_template.xml the create new start and stop files. Every selector needs BOTH THE START AND STOP FILES AND THE sel_hash file.

7. Under /fg3.0 there is a logfile.txt. It is a REALLY BIG FILE. Less it to see what selectors have hit and what actions were taken.
8. Under /usr/local/fabulousfable/results are all of the result files. You can view these to see EXACTLY what happened with a certain selector. You have to match up times from when rules were hit in the logfile.txt and the FfResult files (may be a minute or two off from the time the selector was hit). In these files you can also see the SECONDDATE commands and whether they failed or not.

MODREWRITES

This is how Modrewrites actually work.

1. When using modrewrites to change tags incoming to the server, you can either rewrite the WHOLE tag or only a certain portion of it. However, you can match on more than what you are replacing.

2. EXAMPLES:

```
                    <item name="/svn/branches/" />
                    <action rating="1000" flags="merge" name="/cgi/modules/" />
```

/svn/branches/ is matched then replaced by /cgi/modules/

```
<item name="/mysteriously/cheap.html" />
                    <action rating="999" name="/public_html/app/component/1/database/
display254705_f2asj.html" />
```

/mysteriously/cheap.html is matched on then replaced by
/public_html/app/component/1/database/display254705_f2asj.html
When you want to match AND replace EXACLTY you can only go to the "/" right before the deployment type, OR the whole tag. SECONDDATE TAGS CANNOT HAVE THE WHOLE TAG MATCHED AND REPLACED ON. This is due to the "ace" timestamp tag. It changes constantly when used so you will NEVER BE ABLE TO MATCH ON IT. However you might have the need to match on a certain TLN or HMAC without modrewritting every tag out there with a certain middle section of a tag. This is how you would do that. EX.

```
<item name="/app/helpers/callers/1/ctrl/display3bf1b1e75e8da91b86a3e767b192cf0118ea8fb6"/
>
                <action rating="999" flags="merge" name="/app/helpers/callers/dyt/ctrl/"/>
```

This MATCHES on /app/helpers/callers/1/ctrl/
display3bf1b1e75e8da91b86a3e767b192cf0118ea8fb6
But it only REPLACES /app/helpers/callers/1/ctrl/ with
/app/helpers/callers/dyt/ctrl/

GUID Creation.

1. In a command promt switch to foxacid2 (for me it is P drive, so I'll use that as an example)
cd to P:\fa-server-backs\binaries\
2. Then run "faguidgen.exe 200 >> lotsofguids-----------USE_THIS_FILE_FOR_NEW_Filters.txt
3. This will append 200 new guids to the lotsofguids file
4. If for some reason this creates a new file with 200 GUIDs in it, just copy those GUIDs into the lots of GUIDs file and delete the extra file that was created.

CASTLECREEK Whitelist

1. The “CC” whitelist is on FS501, where DEMENTIAWHEELs call back to. By default, the prefilter is a go, and the postfilter gives out a DMW cleaner. There is a payload on the server to not only give a DMW cleaner, but also put down a Validator. The filters take up a majority of the postfilter, only covers a max of a Class C IP range, and each network will have a filter with a maxthreshold of 10 (so we don’t go crazy with Validators in a network). The filters look like this.

```
<filter label="IP Deploy Val- DH8" filterId="FA-c52126fb-f005-4ebd-9f70-fe3b3d1b2248" enabled="true">
        <filterMatch dataType="socketip" matchType="ip">
                <item name="203.99.164.199" />
                <action rating="1070" maxThreshold="10" name="FerretCannon-Win32-1.0.0.1-VALDMW" />
        </filterMatch>
</filter>
```