**Network Malware Inspection**

* [Maltrail](https://github.com/stamparm/maltrail)

**Data Analytics and Visualization**

* Search: [ElasticSearch](https://www.elastic.co/products/elasticsearch)
* Log Ingest: [Beats](https://www.elastic.co/products/beats), [Logstash](https://www.elastic.co/products/logstash)
* Visualization: [Kibana](https://www.elastic.co/products/kibana), Kibi

**Endpoint Forensic Investigation**

* Open Source: [Google Rapid Response](https://github.com/google/grr), [Kansa](https://github.com/davehull/Kansa)
* Commercial (require additional license): [EnCase Enterprise](https://www.guidancesoftware.com/encase-endpoint-investigator?cmpid=nav_r), [F-Response](https://www.f-response.com/)

**Incident Response Ticketing**

* [Fast Incident Response (F.I.R.)](https://github.com/certsocietegenerale/FIR)

**Log Management**

* [GrayLog](https://www.graylog.org/), syslog

**Examine Browser Malware**

* Website analysis: [Thug](https://github.com/buffer/thug), [mitmproxy](http://mitmproxy.org/), [Network Miner Free Edition](http://www.netresec.com/?page=NetworkMiner), [curl](http://curl.haxx.se/), [Wget](https://www.gnu.org/software/wget/), [Burp Proxy Free Edition](http://portswigger.net/burp/), [Automater](http://www.tekdefense.com/automater/), [pdnstool](https://github.com/chrislee35/passivedns-client), [Tor](https://www.torproject.org/), [tcpextract](http://tcpxtract.sourceforge.net/), [tcpflow](https://github.com/simsong/tcpflow), [passive.py](https://github.com/REMnux/distro/blob/v6/passive.py), [CapTipper](https://github.com/omriher/CapTipper), [yaraPcap.py](https://github.com/kevthehermit/YaraPcap)
* Flash: [xxxswf](http://hooked-on-mnemonics.blogspot.com/2011/12/xxxswfpy.html), [SWF Tools](http://www.swftools.org/), [RABCDAsm](https://github.com/CyberShadow/RABCDAsm), [extract\_swf](https://gist.github.com/noonat/821548), [Flare](http://www.nowrap.de/flare.html)
* Java: [Java Cache IDX Parser](https://github.com/Rurik/Java_IDX_Parser/), [JD-GUI Java Decompiler](http://jd.benow.ca/), [JAD Java Decompiler](http://varaneckas.com/jad), [Javassist](http://www.javassist.org/), [CFR](http://www.benf.org/other/cfr/)
* JavaScript: [Rhino Debugger](https://developer.mozilla.org/en-US/docs/Mozilla/Projects/Rhino/Debugger), [ExtractScripts](http://blog.didierstevens.com/programs/extractscripts/), [SpiderMonkey](https://developer.mozilla.org/en-US/docs/Mozilla/Projects/SpiderMonkey), [V8](https://code.google.com/p/v8/), [JS Beautifier](https://github.com/einars/js-beautify)

**Examine Document Files**

* PDF: [AnalyzePDF](https://github.com/hiddenillusion/AnalyzePDF), [Pdfobjflow](http://www.aldeid.com/wiki/Pdfobjflow), [pdfid](http://blog.didierstevens.com/programs/pdf-tools/), [pdf-parser](http://blog.didierstevens.com/programs/pdf-tools/), [peepdf](http://eternal-todo.com/tools/peepdf-pdf-analysis-tool#releases), [Origami](https://code.google.com/p/origami-pdf/), [PDF X-RAY Lite](https://github.com/9b/pdfxray_lite), [PDFtk](http://www.pdflabs.com/tools/pdftk-the-pdf-toolkit/),[swf\_mastah](http://blog.9bplus.com/snatching-swf-from-pdfs-made-easier/), [qpdf](http://qpdf.sourceforge.net/), [pdfresurrect](https://github.com/enferex/pdfresurrect)
* Microsoft Office: [officeparser](https://github.com/unixfreak0037/officeparser), [pyOLEScanner.py](https://github.com/Evilcry/PythonScripts/raw/master/), [oletools](http://www.decalage.info/python/oletools), [libolecf](https://github.com/libyal/libolecf), [oledump](http://blog.didierstevens.com/programs/oledump-py/), [emldump](https://isc.sans.edu/diary/Malicious+Word+Document+This+Time+The+Maldoc+Is+A+MIME+File/19673/),[MSGConvert](http://www.matijs.net/software/msgconv/), [base64dump.py](http://blog.didierstevens.com/2015/07/05/base64dump-py-version-0-0-1/), [unicode](https://github.com/garabik/unicode)
* Shellcode: [sctest](http://libemu.carnivore.it/), unicode2hex-escaped, unicode2raw, [dism-this](http://hooked-on-mnemonics.blogspot.com/2012/10/dism-thispy.html), [shellcode2exe](https://github.com/MarioVilas/shellcode_tools/blob/master/shellcode2exe.py)

**Extract and Decode Artifacts**

* Deobfuscate: [unXOR](https://github.com/tomchop/unxor/), [XORStrings](http://blog.didierstevens.com/2013/04/15/new-tool-xorstrings/), [ex\_pe\_xor](http://hooked-on-mnemonics.blogspot.com/2014/04/expexorpy.html), [XORSearch](http://blog.didierstevens.com/programs/xorsearch/), [brxor.py](https://github.com/REMnux/distro/blob/v6/brxor.py), [xortool](https://github.com/hellman/xortool), [NoMoreXOR](https://github.com/hiddenillusion/NoMoreXOR),[XORBruteForcer](http://eternal-todo.com/category/bruteforce), [Balbuzard](https://bitbucket.org/decalage/balbuzard/wiki/Home), [FLOSS](https://github.com/fireeye/flare-floss/)
* Extract strings: [strdeobj](http://totalhash.com/download/strdeob.pl.txt), [pestr](http://pev.sourceforge.net/), [strings](http://en.wikipedia.org/wiki/Strings_(Unix))
* Carving: [Foremost](http://foremost.sourceforge.net/), [Scalpel](http://www.forensicswiki.org/wiki/Scalpel), [bulk\_extractor](http://www.forensicswiki.org/wiki/Bulk_extractor), [Hachoir](https://bitbucket.org/haypo/hachoir)

**Handle Network Interactions**

* ;[Wireshark](http://www.wireshark.org/), [ngrep](http://ngrep.sourceforge.net/), [TCPDump](http://www.tcpdump.org/), [tcpick](http://tcpick.sourceforge.net/)
* Services: [FakeDNS](http://code.activestate.com/recipes/491264-mini-fake-dns-server/), [Nginx](http://nginx.org/), [fakeMail](http://sourceforge.net/projects/fakemail/), [Honeyd](http://www.honeyd.org/), [INetSim](http://www.inetsim.org/), [Inspire IRCd](http://www.inspircd.org/), [OpenSSH](http://www.openssh.com/), accept-all-ips, passivedns monitoring
* Miscellaneous network: [prettyping.sh](https://bitbucket.org/denilsonsa/small_scripts/src/3ec16014c839ea0852fae492813ad2293bd61155/prettyping.sh), set-static-ip, renew-dhcp, [Netcat](http://netcat.sourceforge.net/), [EPIC IRC Client](http://www.epicsol.org/),[stunnel](https://www.stunnel.org/), [Just-Metadata](https://github.com/ChrisTruncer/Just-Metadata)
* Deep Packet Inspection: Moloch
* Netflow: FlowBAT, NTOPng, SiLK

**Process Multiple Malware Samples**

* [Maltrieve](https://github.com/technoskald/maltrieve), [Ragpicker](https://code.google.com/p/malware-crawler/), [Viper](https://github.com/botherder/viper), [MASTIFF](https://git.korelogic.com/mastiff.git/), [Density Scout](http://www.cert.at/downloads/software/densityscout_en.html)

**Examine File Properties and Contents**

* Define signatures: [YaraGenerator](https://github.com/Xen0ph0n/YaraGenerator), [IOCextractor](https://github.com/stephenbrannon/IOCextractor), [Autorule](http://joxeankoret.com/blog/2012/04/29/extracting-binary-patterns-in-malware-sets-and-generating-yara-rules/), [Rule Editor](https://github.com/ifontarensky/RuleEditor), [ioc-parser](https://github.com/armbues/ioc_parser)
* Scan: [Yara](http://plusvic.github.io/yara/), [ClamAV](http://www.clamav.net/), [TrID](http://mark0.net/soft-trid-e.html), [ExifTool](http://www.sno.phy.queensu.ca/~phil/exiftool/), [virustotal-submit](http://blog.didierstevens.com/programs/virustotal-tools/), [Disitool](http://blog.didierstevens.com/programs/disitool/)
* Hashes: [nsrllookup](https://github.com/rjhansen/nsrllookup), [Automater](http://www.tekdefense.com/automater/), [Hash Identifier](https://code.google.com/p/hash-identifier/), [totalhash](https://gist.github.com/malc0de/10270150), [ssdeep](http://ssdeep.sourceforge.net/), [virustotal-search](http://blog.didierstevens.com/programs/virustotal-tools/),[VirusTotalApi](https://github.com/doomedraven/VirusTotalApi)

**Investigate Linux Malware**

* System: [Sysdig](http://www.sysdig.org/), [Unhide](http://www.unhide-forensics.info/)
* Disassemble: [Vivisect](http://visi.kenshoto.com/viki/Vivisect), [Udis86](http://udis86.sourceforge.net/), [objdump](http://en.wikipedia.org/wiki/Objdump)
* Debug: [Evan’s Debugger (EDB)](http://codef00.com/projects#debugger), [GNU Project Debugger (GDB)](http://www.sourceware.org/gdb/)
* Trace: [strace](https://sourceforge.net/projects/strace/), [ltrace](http://ltrace.org/)
* Investigate: [Radare 2](https://github.com/radare/radare2), [Pyew](https://code.google.com/p/pyew/), [Bokken](https://inguma.eu/projects/bokken), [m2elf](https://github.com/XlogicX/m2elf), [ELF Parser](http://elfparser.com/)
* Automated Dynamic Analysis: Cuckoo Sandbox

**Edit and View Files**

* Text: [SciTE](http://www.scintilla.org/SciTE.html), [Geany](http://www.geany.org/), [Vim](http://www.vim.org/)
* Images: [feh](http://feh.finalrewind.org/), [ImageMagick](http://www.imagemagick.org/)
* Binary: [wxHexEditor](http://sourceforge.net/projects/wxhexeditor/), [VBinDiff](http://www.cjmweb.net/vbindiff/)
* Documents: [Xpdf](http://www.foolabs.com/xpdf/)

**Examine Memory Snapshots**

* [Volatility Framework](https://github.com/volatilityfoundation/volatility), [findaes](http://jessekornblum.livejournal.com/269749.html), AESKeyFinder, RSAKeyFinder, [VolDiff](https://github.com/REMnux/docs/blob/master/tools/VolDiff.md), [Rekall](http://www.rekall-forensic.com/),[linux\_mem\_diff\_tool](https://github.com/monnappa22/linux_mem_diff_tool)

**Statically Examine PE Files**

* Unpacking: [UPX](http://upx.sourceforge.net/), [Bytehist](https://www.cert.at/downloads/software/bytehist_en.html), [Density Scout](http://www.cert.at/downloads/software/densityscout_en.html), [PackerID](http://handlers.sans.org/jclausing/packerid.py)
* Disassemble: [objdump](http://en.wikipedia.org/wiki/Objdump), [Udis86](http://udis86.sourceforge.net/), [Vivisect](http://visi.kenshoto.com/viki/Vivisect)
* Find anomalies: [Signsrch](http://aluigi.altervista.org/mytoolz.htm), [pescanner](https://code.google.com/p/malwarecookbook/source/browse/trunk/3/8/pescanner.py), [ExeScan](http://securityxploded.com/exe-scan.php), [pev](http://pev.sourceforge.net/), [Peframe](https://github.com/guelfoweb/peframe), [pedump](http://pedump.me/)
* Investigate: [Bokken](https://inguma.eu/projects/bokken), [RATDecoders](https://github.com/kevthehermit/RATDecoders), [Pyew](https://code.google.com/p/pyew/), [readpe.py](https://github.com/crackinglandia/pype32), [PyInstaller Extractor](https://github.com/zrax/pycdc), [DC3-MWCP](https://github.com/Defense-Cyber-Crime-Center/DC3-MWCP)
* Automated: [Cuckoo Sandbox](https://github.com/cuckoosandbox/cuckoo), [Viper Framework](http://viper.li/)

**Investigate Mobile Malware**

* [Androwarn](https://github.com/maaaaz/androwarn), [AndroGuard](https://github.com/androguard/androguard), [Cuckoo Sandbox](https://github.com/cuckoosandbox/cuckoo)

# UHG/Optum Technology System to OAFE Tool Mapping

|  |  |  |  |
| --- | --- | --- | --- |
| Technology | UHG Internal System | OAFE Tools | Notes |
| Deep Packet Inspection | RSA Security Analytics | Moloch |  |
| Network Malware Inspection | FireEye | Maltrail |  |
| Endpoint Forensics | EnCase Enterprise, F-Response | Google Rapid Response, EnCase Enterprise, F-Response | EnCase and F-Response require valid licenses |
| Incident Response Ticketing | RiskVision | Fast Incident Response | This requires consulting time at an additional fee. |
| Data  Analytics and Visualization | Security Big Data Lake, Elasticsearch, Logstash, Kibana | Elasticsearch, Logstash, and Kibana | Additional consulting fees are required for configurations that involve anything more than basic search functionality for CFI Investigators. |
| Log Management | HP ArcSight, Security Big Data Lake | Graylog (ELK Stack required) | This is enabled on a fee basis. Monitoring is an additional charge. |
| Malware – Dynamic Analysis | ThreatAnalyzer, Cuckoo Sandbox | Cuckoo Sandbox |  |

# OAFE Application Ports

|  |  |  |  |
| --- | --- | --- | --- |
| Application | Local access (loopback) | Internal Access | Notes |
| Cuckoo Sandbox Web Interface | 8511 (tcp) | 8501 (tcp) | Proxied via nginx. Other languages can be configured, if required. |
| Cuckoo Sandbox API | 8510 (tcp) | 8504 (tcp) |  |
| Maltrail | 8338 (tcp) | 8338 (tcp) | Internal authentication provided by Maltrail |
| Google Rapid Response | 8000 (tcp) | 8000 (tcp), 8080 (tcp – agent access) | Agents access the server on port 8080. Port 8000 is management interface. |
| Kibana | 5601 (tcp) | 8502 (tcp) | Proxied via nginx |
| Elasticsearch | 9200 (tcp) | N/A | Can be configured through proxy. |
| NTOPng | 3000 (tcp) | 3000 (tcp) | Internal authentication |
| Graylog | 9000 (tcp) | 9000 (tcp), 12900 (tcp) | Web console is accessed via port 9000 |
| Viper | 8514 (tcp) | 8505 (tcp) |  |
| Kibi | 5606 (tcp) | 8507 (tcp) |  |
| Moloch (DPI) | 8005 (tcp) | 8005 (tcp) | Internal authentication |
| Logstash | 5004 (tcp) | N/A |  |

|  |  |
| --- | --- |
|  |  |

**O.A.F.E. Pre-Install Checklist**

**Site Location**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Company Name: |  |  | Onsite Contact: |  |
| Address of install: |  |  | Project Contact: |  |

**Network**

|  |
| --- |
| Configure and cable two (2) monitoring ports (SPAN, port-mirror, TAP, Gigamon, etc). 1 Gigabit minimum. Please consult CFI staff about 10 gigabit monitoring.  Assign IP address on internal network and cable 1 port for management and communication to the internal network.  Provide IP address information to CFI consultant. Please include IP address, subnet mask, gateway, DNS servers, and WINS servers (if available).  IP address and cabling for iLo (remote server management) device. Please include IP address, subnet mask, gateway, and DNS server information.  The CFI consultant will need access to the internal network to configure the device. This access can be wired or wireless, but should be able to access the server. |

**Firewall Rules**

|  |  |  |
| --- | --- | --- |
| Configure the firewall | * VNC – Allow VNC from VPN clients * SSH – Allow SSH from VPN clients * Allow outbound traffic from management interface directly to the internet (for install and updates). No proxy. * Allow outbound OpenVPN (udp 1194) from management interface to CFI External IP range |  |

**Remote Network Access (VPN, Firewall, etc)**

|  |  |  |
| --- | --- | --- |
| Provide CFI with remote access to the OAFE device | * VPN access (if available). CFI will provide a list of users and email addresses. * Firewall rules (inbound firewall rules can be created if VPN access is unavailable) * Outbound VPN (OpenVPN can be utilized to access the device, if required) * Site to site VPN (This can be configured on an as needed basis. Please consult with the CFI consultant assigned to the project) |  |

**Instructions for CFI consultant**

|  |  |
| --- | --- |
| Please provide the CFI analyst with any information pertaining to locating the facility, how to access the facility, and any other special instructions. | |
| Guest wireless access. |

**Physical and power requirement**

|  |
| --- |
| The Optum OAFE device requires 2 U of Rackspace in a standard server rack.  Two (2) power (NEMA) connections are required and should be connected to a UPS.  Ensure all network cabling has been completed. There should be four (4) cables. Two (2) cables for management and two (2) for monitoring. Cables should be properly labeled. |