Ransomware 1st Response Run-book

## What is a Ransomware ?

Ransomware is a type of malware from crypto virology that threatens to publish the victim's personal data or permanently block access to it unless a ransom is paid off.

Possible surfaces via which a ransomware or any malicious payload can enter:

### External

The attacker gradually enumerates the underlying infrastructure by spidering the publicly available corporate webpages and eventually applications

* POST webpages with possible SQL injection and leading to arbitrary Remote Code Execution, this leads to directly database compromise and possible leak or encryption.
* Exposed admin pages to public network are a good playfield for brute force attacks to find and exploit admin credentials, for this attacker usually depend upon globally available password/username wordlists
* Upload links allowing to upload images / documents without implementing a scanning mechanism leading to RFI / LFI attacks
* Publicly disclosed Info pages like – website.com/phpinfo allows the attacker to view web-server version information, this help to isolate exploits as per available CVE’s.
* E-mail phishing links / attachments that may come in pdf / word format, there are payloads cloaked behind these attachments using a technique called steganography.
* Public facing application FQDNs without WAF covering OWASP Top 10 or public facing servers with management ports or anonymous ports open, allowing the attacked to sniff the server using various tools and techniques.
* Loose firewall rules with improper SSL-inspection & IDS/IPS controls defined

### Internal

The attacker has already gained access to a server / user-account and is traversing in the network finding maximum reachability and mapping the architecture to plan a distributed attack

* User-Accounts with un-scoped roles
* Zombie Service-Accounts
* Unpatched version of software / OS with existing CVEs (**Refer: Article A**)
* Cron-jobs / schedulers being run by user accounts can be used to run payloads
* Application code being kept in an unsecured repository / server
* ACLs over file-shares & allowed commands with NOPASSWD ( For Linux distributions ) are major causes for RCEs and attacks via payload injection
* Un-restricted privileges for normal domain users to be able to query the system via CMD

## Recovery & Defence Plan

### Isolate

* Once effected, cut-off networking of the victim machine, goal is to prevent further traversal. Default payload behaviour is to find and target database servers and then move on to the remaining infrastructure.
* Shutdown the victim server and create a copy of the disk and send it to a sandbox / forensic environment to work on how the payload entered
* Make sure you project maintenance window on your site to avoid any end user data incoming and stop your database services to avoid further encryption
* Consider all the application servers or any other machine with possible reachability to the victim server as “Affected”

### No Immediate Restores

* Do not immediately restore from latest backup point, choose a backup point of a week later, some payloads are time sensitive and will trigger automatically even after you restore as per their programmed schedule
* If you have backup solutions that keep your backed up data in an encrypted manner, use them to restore from a week-old point.
* Apply backup image scanning mechanism using tools available in market. (**Refer: Article B**)
* Your backup mechanisms should be smartly scheduled, a backup job should run only after a deep malware scan has completed with 0 detections, if anomalies are detected, backups should not get triggered.
* Keep a second layer of backup on a different destination, example, primary backup and sit on VEEAM for on-premises, take a secondary encrypted backup on Cloud.

### Review & Modify Entry Points

* While your data is being restore in Step#2, begin the exercise for identifying all the application FQDNs and DNS records for the same.
* Change the FQDNs by either re-direction or proxy or consider re-naming with a slight change, make sure the resolution IP is changed as well.
* Make sure all your L7 is routed through a very aggressive WAF solution. (**Refer: Article C**)
* Entry points to your database should be strictly restricted and key-based authentication should be practiced for service accounts / user accounts
* Consider leveraging least access model for all user within a database, for example, anyone should not be admin for entire databases. However, for use-cases access can be crated and revoked post requirement is fulfilled

### Nano-Segmentation

* Not every network should talk to every other network
* Each application and database subnet should be restricted to talk over required ports by allowed servers only, no anonymous access on any port should be allowed.
* A default DENY-ALL rule should be practiced in any network
* Create separate application LAN segments and their respective Database segments as well, avoid using a shared database model, considering each application has its own vulnerabilities and can compromise the entire database if being used for critical applications as well.
* Restore the servers once above structure is ready accordingly to their required LAN segments

### Scan & Validate

* Initiate an aggressive scan using EDR | MDR solutions (**Refer: Article D**)
* Ensure you have real-time memory scans going on, this will impact server performance but will ensure no backdoors are running behind system processes
* Depth scans should run twice – Before business hours & After Business hours on daily basis
* Use Vulnerability Assessment scanners to ensure you’re safe from existing CVEs (**Refer: Article E**)
* Run Password Audits and Active Directory X-rays (**Refer: Article F**)
* Based on reports, mitigate anomalies If everything looks good, Go Live

Audit & Logging

* Practice Bi-annual VAPT on all staging and production machines
* Leverage a SIEM / SOAR solution for all components for your infrastructure (**Refer: Article G**)

# Reference Documents

* **Article A**: Repositories to see available exploits for various platforms
  + <https://www.exploit-db.com/>
* **Article B**: Scan your backups before restore
  + <https://www.veeam.com/blog/how-to-scan-backups-for-ransomware.html>
  + [https://helpcenter.veeam.com/docs/backup/vsphere/surebackup\_hiw.html?ver=110&\_gl=1\*15hrcit\*\_ga\*NjM4NDc0OTg0LjE2NzMwMDMxOTQ.\*\_ga\_PMJS81E58L\*MTY3MzAwMzE5OC4xLjAuMTY3MzAwMzE5OC42MC4wLjA](https://helpcenter.veeam.com/docs/backup/vsphere/surebackup_hiw.html?ver=110&_gl=1*15hrcit*_ga*NjM4NDc0OTg0LjE2NzMwMDMxOTQ.*_ga_PMJS81E58L*MTY3MzAwMzE5OC4xLjAuMTY3MzAwMzE5OC42MC4wLjA)
* **Article C**:
  + <https://www.cloudflare.com/waf/>
  + <https://www.f5.com/pdf/solution-overview/f5-big-ip-advanced-waf-protection-for-every-app-anywhere.pdf>
  + <https://www.barracuda.com/products/application-cloud-security/web-application-firewall/features#paranav-navbar>
* **Article D**:
  + <https://www.sentinelone.com/surfaces/endpoint/>
  + <https://www.crowdstrike.com/products/endpoint-security/>
  + <https://www.trendmicro.com/en_in/business/products/detection-response/xdr.html>
* **Article E**:
  + <https://www.qualys.com/apps/vulnerability-management-detection-response/>
* **Article F**:
  + <https://github.com/ClaudioMerola/ADxRay>
  + <https://mortenknudsen.net/?p=403>
* **Article G**:
  + <https://azure.microsoft.com/en-us/products/microsoft-sentinel/#overview>