Meri Kann cibR punX
unix network kernel

t.(o0)_j
SKRP

journey to the land of unix
experience C, shell & perl

ways of deep sea secrets
raw power of binary-logic

this ordeal reward 1337
hard for all way wizards

master one machine
masters all machines

reign of wizard calcT
are you ready to go?

##

This work describes & details a unix network built with the mission to create anonymous machines which network together to allow file sharing, communication and secured connection.

The nature of both the machine node & the network are disposable. This nature keeps the footprint low which enhances the anonymity of the user. Nodes are quickly wiped.
Nodes have a protocol to be introduced into the Network which allows for automated cohesion that enhances the durability of each network.

Contained is the clear and simple process of propagation from code to a fully functioning network.

The network is entirely decentralized.
Every connection is peer to peer enshrouded by encrypted tunnels.

Services offered:
Encrypted local & remote command line terminal interface to unix kernels.
Environment & space to code & compile.
Massive data share & storage.
Clan user groups for permanent groups & file space.
Web https servers & clients.
File transfer via secure-file-copy (scp).
Email clients & servers (internal to network & external to internet).
IRC clients & servers (internal to network & external to internet).
Proxy actions thru a unique string of nodes. Each secured by common protocol.
Various other services may be offered such as games.

##

Preserve all insight
Till the last man walk

That he might not walk
In the darkness of past

Independent of
Time wealth or government

##
Contents
##

##
SICC System In Complete Chaos File System
##

MISSION

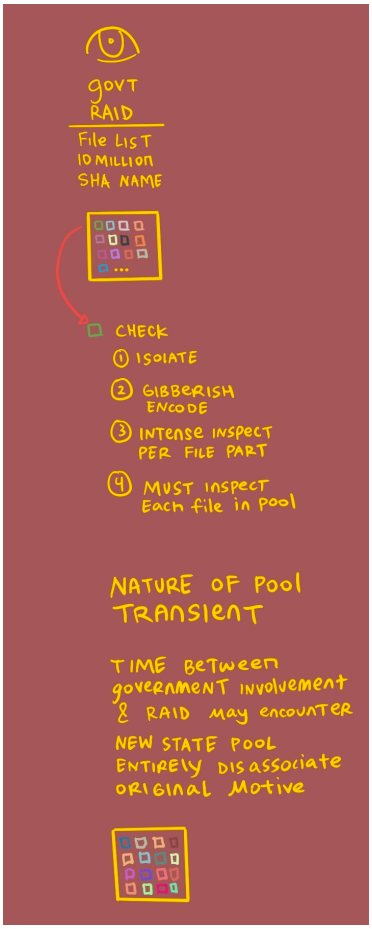
System in Complete Chaos uses obfuscation to establish a network file system that a user with access keys to a node may obtain via encrypted tunnels.
To obtain a specific file there are three necessary components:
  Metadata which contains filename, encoding, size, and archive index.
  Data parts associate with the file
  FKEY which is the sequence of parts required to build a complete file.

DATA

The heart of the system is in the wealth of data.
The unix kernel and network implementations center around data archiving.
Each file is split into parts and stored as unassociated parts spread over the network pool.
Each part of the file is then isolated from the whole.
This sets all file parts equal and unbiased.
To preserve any file the entire network pool must be preserved.
The mission is to keep data public yet to isolate the individual node from responsibility of the data hosted.

LIABILITY

The purpose of the system to to isolate the operator of the node from the data hosted on the node. The entire pool contains only parts of files that are unknown & unassociated to the operator of the node.
There is no association if exist more than one part of the same file.
The operator then is less liable for the data hosted. The operator may choose to implement protocols to have the system manage the data present at any time.
The operator may set a pool refresh cycle. This keeps the data moving which narrows the window of action that a government can access the data on the specific node.



METHOD
Each file is processed by a standard protocols.
The file is split into parts and propagated into a pool of data which exists across the network.
WORLD is the only place that the file exists whole is in a secured private user community and in offline backup drives.
FKEY is the sequence of parts that compile the file. This maps a file to its various isolate parts. Without this sequence a file has no method to be built.
FKEY is transferred via usb to a designated archive location. This is done via an offline-usb transfer to keep the critical component immune to interception or snoops.

NODE
Each node may have entire control over the size of the pool of data offered and also the files offered by the node.
Each node carries a partial section of the network pool. This entire section only stores disassociated file parts which were split across the network.
The node periodically reports the contents of the pool which users compile into a file census to determine level of duplication and maps the file part to the set of nodes which host it.
Nodes may offer a WORLD file space which host complete files. The compiled file space is entirely isolated from the file parts of the network pool.
Creation of a node, a kingdom to the glory of the wizard.

NETWORK
CENSUS keeps a live accounting of the pool contents of each node compiled into a grand map.
Users access a CENSUS node to locate the set of nodes which contain the file part.
META node keeps a database associating: filename, encoding, size, and archive index. Users search this database to map the complete file digest to the file FKEY, which contains the sequence to build the file.
BASTION node uses offline-only usb transfers of FKEY to obtain the sequence of a file in a way that adds a layer of protection to the critical nature of the FKEY.

SEED
BACKUP off-line backup drives contain complete compiled files.
In event of propagation or network corruption or network dissolution; the entire pool can be rebuilt using the backup drives to seed a new network pool of data.
BACKUP drives get connected to a node then the files are split into parts. Those parts are then distributed to various nodes to restore a network pool.

PROCESS
Extract files from an external non-network source.
Slice the file into parts.
Transfer the parts to various nodes via obfuscation protocols.
Verify integrity of file part.
Build file from a sequence of parts, each part found on different nodes.

CENSUS
Each node has a file pool that can be listed & verified by any user with access.
This adds an independent check to archive integrity.
File pool lists are compiled into a network list. This is compiled into a network wide map of file part to all the nodes which host that file part.
Duplication is calculated and then users can duplicate files as required.
Duplication on this system is always explicit & controlled. This method keeps the archive at maximum efficiency.

FILE
File parts are named after their unique file digest. This is a mathematical algorithm that ensures, based on the binary content, each unique file part has a unique file name.
If the file digest is computed and it does not match the name then corruption is immediately detected.
In this naming system, duplicate files are immediately detected because the digest will be the same. Hence, the system will detect duplications naturally.

META
Metadata is isolated from files and FKEY. Metadata is critical to aide in finding files.
Metadata information: filename, size, encoding, clan, source-path, INDEX.
INDEX is the archive system used for grouping & associating files.
Detail of INDEX is what gives the database usefulness. Book:genre:subgroup:author:edition
Users are incentivized to process & rate the file, which will benefit the CLAN that uploaded the files. Trash files reflect back upon the CLAN uploader.
META databases are on secure bastions. Users are able to search to obtain files. The search returns the complete file digest. Next, the users need to obtain the FKEY for the associated complete file digest. Then, the user obtains the file parts from various sources. Last, the file is compiled & verified.

POOL
Many different file pools can exist.
Known metadata only files ensure that the file has been processed and is acceptable to a group.
Massive obfuscation which compiles many different pools with intent for higher security.

TOOLS
Extract: recursive file copy of a target device which parses metadata & dumps data into a standardized pool. Data is by nature de-duplicated and such instances are echoed in the metadata file.
Scrub: verify file digest matches the standard filename, which is named after the digest.
ChkMeta: confirm each file in SICC pool has metadata.
UNIQ: compare two linked-lists to output all non-duplications.
XFR: copy massive amount of iterations to a storage-drive.
INDEX: from metadata compiled build a searchable database and edit archive index of files.
LONE: searches for non-duplicated files in an archive to create a BOUNTY list of files for nodes to propagate.

Mathematic Structures
SICC C code: Extract, Slice, Verify, Build
Network C code: scp, propagate
MKRX C code: Census, Auto_Pool_Cycle, Metadata, Bounty

##
MKRX unix network kernel
##

INTRO
The archive exists in the unix universe.
Each node is an entity.
Nodes which are synchronized exist as a network.
Protocol establishes initialization into the network.

KERNEL
Each node is a homogenized unix kernel. Every environment is a FreeBSD World.
Skill & techniques developed upon any one will exactly carry over to any other machine.
Command Line is method which users interact with the unix kernel.
Machines may be different but the methods to interact with each is the same.
Kernels may be optimized with custom programming, yet the environment is the same.

CLONE
Power of the unix kernel is its ability to map its system atop various machines in a homogenous environment.
Each node is disposable, as installation is a trivial matter with physical access to the machine.
If a node loses synchronization with the network then the node must be wiped.
All access keys are nullified and permanent environments may be copied onto a new installation.
New installations must rebuild using a new name.

JAIL
Users interact only with a sandbox image of the main-host system called a FreeBSD Jail.
Jails provide a working environment that is disposable.
Jails also aid security & stability by limiting scope of users upon the main-host system.
User-space exists in the jail, but it is managed by the host system.
The user-space is periodically scrubbed once the account it used to leave-no-trace.

$KEY
$KEY are access-keys to a node. The access is restricted to the jail. The sandbox layer of the node. This isolation adds to the stability of the host machine.
All access keys are one-time-only usage. If the key is a victim of a snoop or copy then it can not be reused.
Disconnections will end the use of the $KEY.
Host machine automates a management of $KEY with a bank of accumulated $KEY of own jail & jails of other machines.
Users manage $KEY by a PACK which it builds locally and which it forwards with each jump.

NODE
Node host machine is the actual kernel.
All access to the host machine are by OPIE access. One Password In Everything.
Clan of the node is responsible for access to the main system.

The node is configured during install.

ROOT
Root to the system is the path to have total control over the machine & jail.
Clan user must ssh using a special account called HEIR user. Which externally appears to be a normal jail user. HEIR requires special authentication protocols.
After authentication the HEIR is able to ssh into the host system.
Next the host ssh account su, switch user, into the unix WHEEL OPIE.
Last the WHEEL user can su into root user.
WHEEL & root shell initialization trip a network wide alert. Root access is discouraged.

CLAN
Jail users are temporary. After access is terminated the account is wiped.
In order to achieve persistent file space the user must su (switch user) into a CLAN.
Each node hosts a jail group for each CLAN in the network.
CLAN may host WORLD which allows anonymous users to have persistent file space.
Thermal-printers are used to aide OPIE (One Password In Everything). This protocol requires a new password with each login. The password is sequential and thus impervious to replay attacks. The sequential nature also informs the user how many times the account has been accessed. If the password sequence is ‘x’ numbers ahead, then the account has been compromised.
There is a password for each iteration of the sequence. This list is printed out using the thermal printer.
Host nodes are transient by nature. CLAN is the structure that builds permanence in the network.

DAY
MKRX unix network operates in cycles of 100 hours.
The decentralized network protocol requires each node to independently verify every other node in the system using OTOPSK (One Time Only PreShared Key).
This reverification assures that all systems are indeed who they claim to be.
Host machines stop the Jail, wipe all used accounts and new accounts are created.
A total reset of the jail system and all network traffic.
The jail is restarted and the reinitialization to the network begins.
REP for each node is sychrononized with the git protocols.

VAULT
Git is used to synchronize all nodes into harmony decentralized.
The main directory lists all the days of the network.
Nodes created after then require the previous days in order to initialize into the network.
Each day has a REP file which is a table CLAN:node-REP:total-REP.
Each day then has a motd file for every node which was consistent in the network.
Motd file acts as the node banner to the network. CLAN:SICC-size:Services:Uptime
The last DAY allows all users who have access to any node a MAP of the network.
Users are able to browse this read-only directory tree to find which nodes to find $KEY access.

REP
Each node is claimed by a CLAN user group.
Every cycle the nodes of the network are valued.
REP is a summation of the value of all nodes of the CLAN which remained consistent.
REP is critical in determining $KEY trade rates. Higher REP purchases $KEYS of lower REP at substantial discounts.

This system allow monetization of a finite, controlled supply with value. The CLAN is incentivized to node betterment to obtain better trade rates.
$KEY are traded or sold using offline-USB transfers via hustlers.
CENSUS also adds rep to CLAN for first-file-scrapes.

HIVE
Ever-present host entities exist in the jail system to interact with users.
Buy/sell $KEY, file space quotas.
Reward quests: log witness, SICC duplications, file scraping, website scraping, index files…

ANONYMOUS
Host are able to communicate mainly via HIVE daemons.
All internode user traffic is via jail sandbox accounts. Isolation of user traffic keeps kernel stability in check.
HIVE daemons are able to keep stable host-to-host communications and terminate all user traffic by shutting down the jail.

LOG
System logs
Dtrace

Norm user
Log, system, dtrace
Directory Hierarchy