
SECURITY GUIDELINES

- ▶ HUBzero tutorials
-

INTRODUCTION

- ▶ Host: Erich Huebner, Science Gateway Manager
- ▶ Guest: Pascal Meunier, Senior Software Engineer
- ▶ Agenda:
 - ▶ HUBzero's proactive approach to security
 - ▶ How Open-Source community members can harden their machines
 - ▶ Questions & Answers

What's Security?

- Can be intimidating, chore, obstacle
- Basic security prevents harm
- Great security allows you to focus on what you need to do instead of fending off problems and wasting time, or do something new

Business Case for Security

- Risks
 - What can you lose if you don't do it
- Perennial questions:
 - What are your assets and provided value?
 - What are the obligations and expectations?
 - What do you want security to enable you to do?
 - How much safety does X provide and how much does it cost?
 - What is most worth doing?
 - How do you measure the benefits?

What are we doing here?

- Education
 - “Workforce hardening”
 - Learn about HUBzero security features
 - Learn about server hardening
- Due Care and Due Diligence
 - Understanding your supplier's (HUBzero) security posture
 - Best practices
 - Bravo for checking ours by watching this

Where does security happen?

- Security can't exist on its own, as a separate process
- Security is part of:
 - Operations
 - Customer service
 - Software development
 - Research
 - Management

HUBzero's Approach to Security

- Like everyone else:
 - Monitoring and Awareness
 - Detection
 - Reaction
 - Prevention
 - Planning and Policies
 - Procedures
 - Education
 - Audits
- HUBzero: Emphasis on time and tailored security

Tailored, Practical Security

- Effective things that save time
 - Scans
 - Benchmarks
 - Software updates
- Apply everything you can from that advice
- Stop when inconvenience or cost mounts
 - Find other approaches or mitigations, discuss
- Architectural Techniques
 - Compartmentalization
 - Layers

Not Just Technical

	Operations	Customer Service	Software Dev	Research	Management
Monitoring	X	X	X		
Awareness	X	X	X		X
Detection	X	X	X	X	
Reaction	X	X	X	X	
Prevention	X	X	X	X	X
Planning & Policies	X	X	X	X	X
Procedures	X	X	X	X	X
Education	X	X	X	X	X
Audits	X	X	X	X	X

Everyone can contribute to security, and everyone is responsible for their part

Criticality of Time

- Time defeats passive security
- Anything connected is constantly under attack
- Need to change or react before attackers succeed
 - Passwords, keys, secrets
 - Protect or change software (apply patches) before vulnerabilities are exploited
 - Ban attackers
- Attackers are incredibly fast
 - You have very little time to apply software patches
 - Some vulnerabilities get exploited even before there are patches

OODA Loop Concept

- Observe, Orient, Decide, Act
- War: Fastest correct OODA loop tends to win
- Delays:
 - Increase security risks
 - Reduce QA risks by increasing quality assurance
- Software updates:
 - Excellent QA by Debian, RedHat
 - Not much more we can do, and when there's a problem it's usually quickly obvious
- HUBzero
 - Non-production gets patched automatically
 - Production gets patched one day later

What You Can Do

- Frequent software Updates
 - Manually every day, or
 - Packages "yum-cron" or "unattended-upgrades"
 - Automated installs
 - Ideal for development machines or workstations
 - May be for production if you don't have the time or budget to worry about this
 - At a minimum, check for available updates as often as you can
- Reboot after kernel updates, whenever possible

Monitoring

- Humans cannot monitor things in real time
 - Need automated subsystems to
 - React, perhaps ban
 - Point out what's unusual
 - Analyze and produce summaries
- Example: System resources and utilization
 - Is a partition filling up quickly? Why?
 - Capacity planning issue?
 - Attack?
 - Accident?
 - Malfunction?
 - You want to know before something stops working

Detection

- Review activity
 - Build a profile of what is normal or does not need attention
 - Review what's left as it happens
 - Unexpected changes?
- Scans
 - Periodic
- Benchmarks
 - Configuration vs accepted good practices

What We Do, and You Can Too

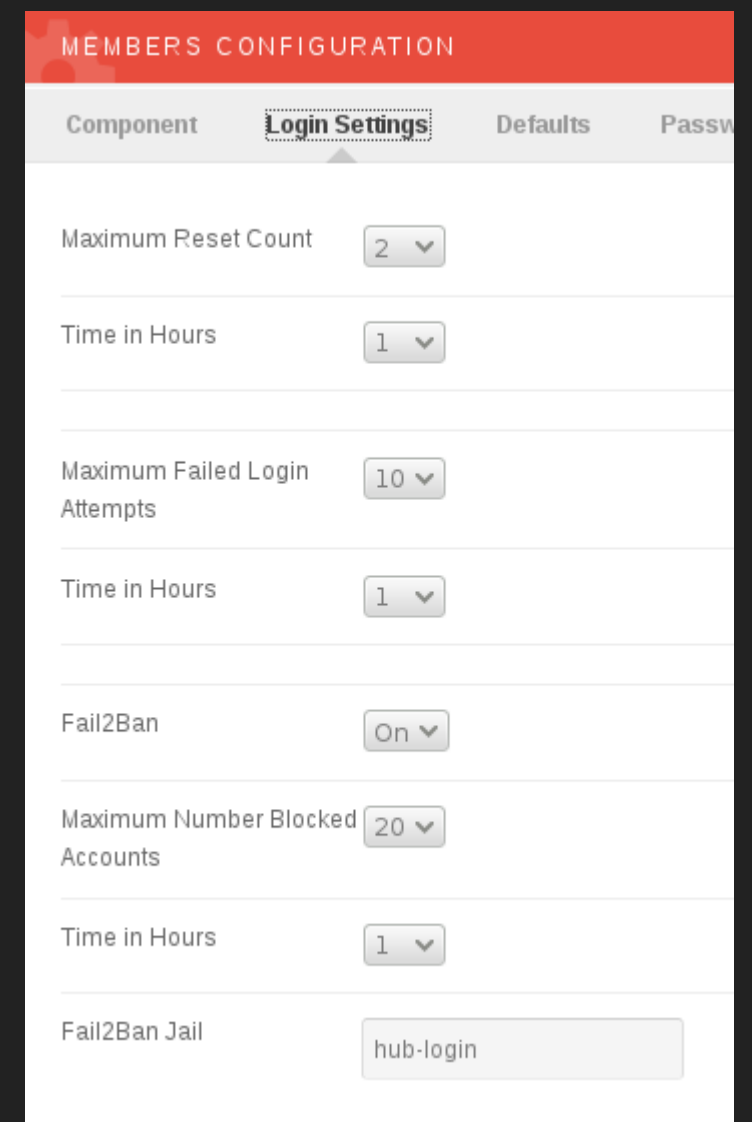
- Use a monitoring system
 - Get it as a service to save work
 - Xymon, Nagios, CloudWatch
- Logs
 - Too much information! Need a log monitor
 - We use Logcheck to filter routine messages and things we don't care about
 - Read the rest
 - Downside: maintaining regular expressions

Most Important Things To Do

- Install fail2ban or equivalent
 - Install and forget (mostly)
 - To defend against brute-force attacks
- Use HUBzero software-managed bans
 - Per {user, IP} pair, not just per IP
 - Much better than fail2ban alone:
 - Won't ban a common wireless access point
 - Unless it's a last resort

CONFIGURE YOUR BANS

- ▶ Password resets/hour (2)
- ▶ Ban {user, IP} after 10 failures/hr
- ▶ Leverage Fail2ban
- ▶ Ban IP after banning 20 {user,IP} pairs/hr
- ▶ Need a recent version of fail2ban
 - e.g., 0.96

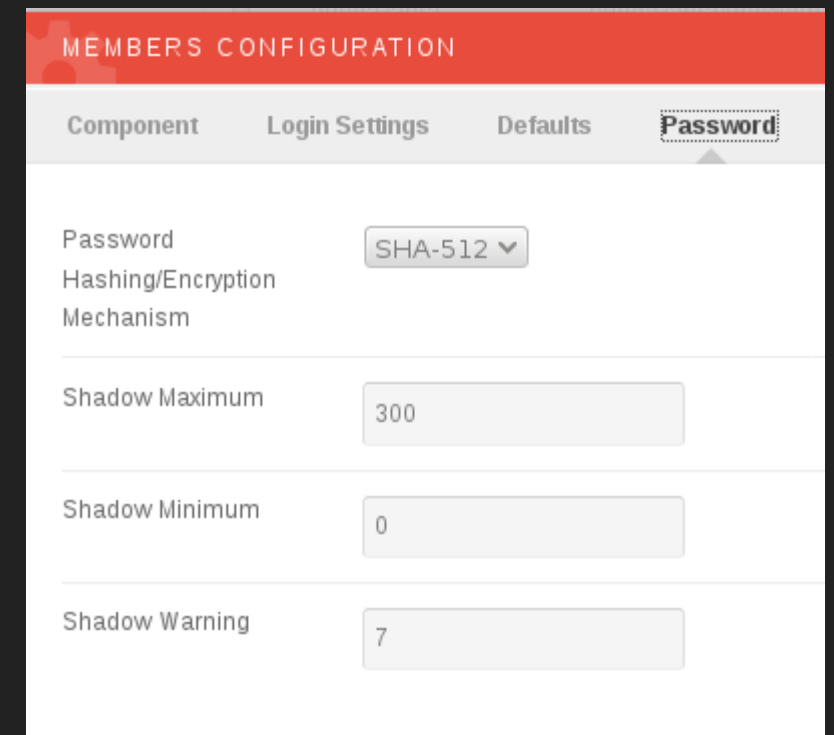


The screenshot displays the 'MEMBERS CONFIGURATION' interface with a red header. Below the header, there are three tabs: 'Component', 'Login Settings' (which is selected and highlighted with a red border), and 'Defaults'. The 'Login Settings' tab contains several configuration options, each with a dropdown menu:

- Maximum Reset Count: 2
- Time in Hours: 1
- Maximum Failed Login Attempts: 10
- Time in Hours: 1
- Fail2Ban: On
- Maximum Number Blocked Accounts: 20
- Time in Hours: 1
- Fail2Ban Jail: hub-login

CONFIGURE YOUR PASSWORDS

- ▶ Use strong hash
- ▶ Expire passwords depending
- ▶ on the password strength you
- ▶ require
- ▶ Warn a few days before expiration



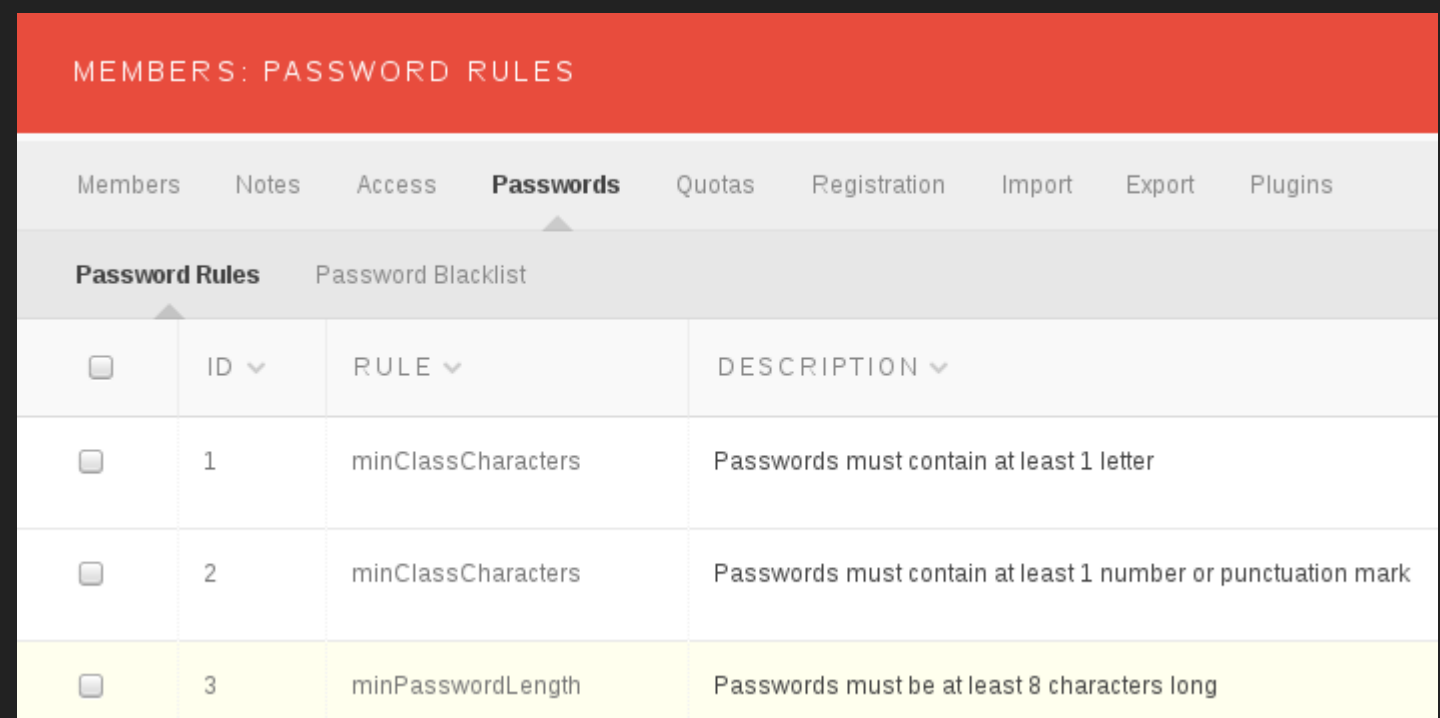
The screenshot displays the 'MEMBERS CONFIGURATION' interface with a red header. Below the header, there are four tabs: 'Component', 'Login Settings', 'Defaults', and 'Password'. The 'Password' tab is selected and highlighted. The configuration area contains four rows of settings:

Setting	Value
Password Hashing/Encryption Mechanism	SHA-512
Shadow Maximum	300
Shadow Minimum	0
Shadow Warning	7

SET PASSWORD RULES

- ▶ Avoid easily guessed, short passwords
 - Minimum length (e.g., 8)
 - Characters of each type (numbers, letters)
 - Not based on user name
 - Different from last

- ▶ Use 2FA with other
- ▶ login methods
- ▶ (e.g., Google,
- ▶ Shibboleth)

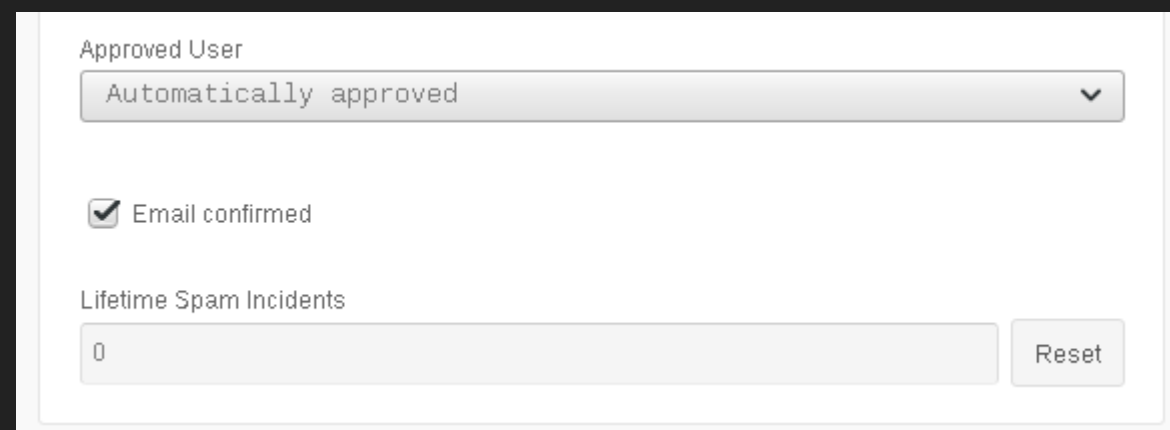
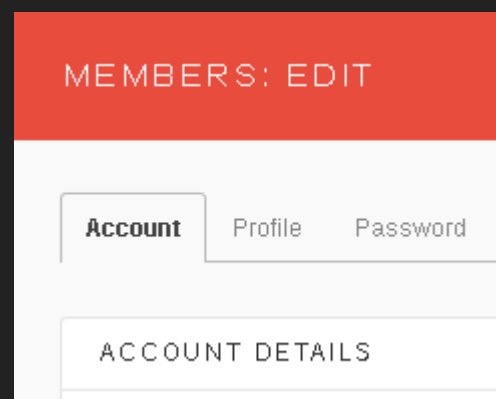
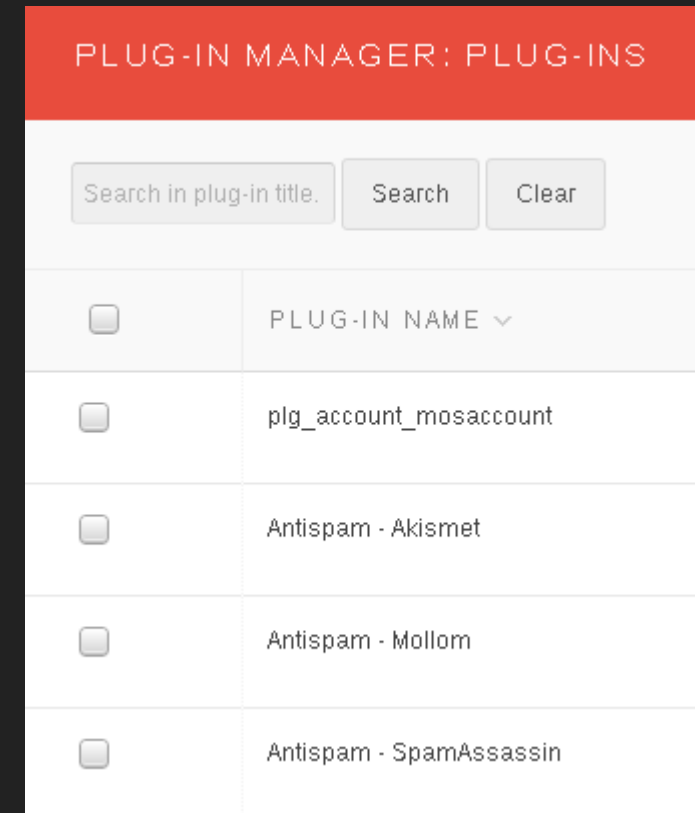


The screenshot shows the 'MEMBERS: PASSWORD RULES' interface. It features a navigation bar with tabs for 'Members', 'Notes', 'Access', 'Passwords', 'Quotas', 'Registration', 'Import', 'Export', and 'Plugins'. Below the navigation bar, there are two sub-sections: 'Password Rules' and 'Password Blacklist'. The 'Password Rules' section contains a table with the following data:

<input type="checkbox"/>	ID ▾	RULE ▾	DESCRIPTION ▾
<input type="checkbox"/>	1	minClassCharacters	Passwords must contain at least 1 letter
<input type="checkbox"/>	2	minClassCharacters	Passwords must contain at least 1 number or punctuation mark
<input type="checkbox"/>	3	minPasswordLength	Passwords must be at least 8 characters long

SPAM PROTECTION

- ▶ Antispam plugins
- ▶ Spam incidents are counted
- ▶ After 10 incidents content
- ▶ submission is blocked
- ▶ Reset count manually to re-enable content submission



SSH KEYS

- ▶ SSH keys are like passwords
 - They can be leaked
 - They can be brute-forced
 - Recommended key length keeps increasing
- ▶ Change SSH keys when you move to a different machine
- ▶ Restrict their validity with "from" option in the `authorized_keys` file
 - Mitigates leaks or theft

SECURITY TRAINING

- If you program PHP, read the PHP Security Cheat Sheet
 - https://www.owasp.org/index.php/PHP_Security_Cheat_Sheet
- ▶ Share news articles
- Phishing: “GitHub Repository Owners Targeted By Data-Stealing Malware - Slashdot”
- ▶ Vulnerability scan reports can be very instructional
- ▶ Conference presentations

SCAN TYPES

- Change control
 - Daily or more
 - Find out what changed on the system
 - Find out status of packages
 - Check for world-writable files, etc
- Configuration
 - When installing new software and at least yearly
 - Compare to good practice
- Vulnerability
 - How does it all behave?

EASY EXAMPLES

- Change control
 - Rkhunter
 - “Warning: Changes found in the group file for group 'www-data':
 - User 'myfriend' has been added to the group”
 - Is that OK?
 - “dpkg - - audit”, “rpm -V”, “yum verify”
 - Failed package update: “The following packages have been unpacked but not yet configured.”

CONFIGURATION

- Lynis
 - Gives an overall “hardening index” in report
 - e.g., “Hardening index : [76]” (out of 100)
 - Free, easy to use
 - Suggestions, e.g., “One or more sysctl values differ from the scan profile”
 - Some tests are simplistic and just look for keywords
 - “SSH option AllowTcpForwarding is in a weak configuration state”:
 - Not taking into account Match clauses

Compartmentalization and Layers

- Firewalls - Why so many types?
 - Network
 - Know most about protocols and host types
 - Host
 - Know most about services
 - Application
 - Know most about users and contextual meaning of data
- Multiple layers resist defeat of one
- "Better" decisions at some layers
- HUBzero deployed network and host firewalls, and bans at the application level

HUBzero Host Firewall

- HUBzero firewall package strengthened
- Goal of first version: allow tools to work
 - Setup NAT
 - Setup forwarding
 - Open firewall, you needed to add rules to secure it
- Now has a DROP policy default
 - Allows only traffic that may be needed for a hub
 - Much more secure by default
 - Needs final tweaks
 - Trim rules that are not needed in your case
 - Add rules for extra services or exceptions you allow
-

Compartmentalization and Layers

- Compartmentalization also means separating databases and services to avoid catastrophe
 - HUBzero has some support for it, which you can use if you want (more work)
- Use different users for different things
- Separate code into directories with different permissions
 - HUBzero has support for this (not described in this presentation)

DISASTER RECOVERY DRILLS

- ▶ Untested backups are likely no backups!
 - Simplest test: restore a file at random
- ▶ FISMA, HIPAA:
 - Come up with a fake exercise such as the loss of the host HUB and requirement to complete restoration from backups
 - Determine participants and select a random date to run the drill
 - Document effective and ineffective areas and retain the report as historical information

SENSITIVE DATA?

- ▶ Inventory your systems and make sure they meet portability and accountability restrictions, such as HIPAA rules
 - ▶ HIPAA: Health Insurance Portability and Accountability Act of 1996
- ▶ Identify and prioritize the risks of the confidentiality
- ▶ Identify weaknesses and overall risk posed

EXAMPLE SENSITIVE DATA

- ▶ SSH `authorized_keys` file contains public keys
 - What could be the problem with **public keys**?
- ▶ On a network share
 - IP address protection only (NFSv3)
- ▶ User has `sudo` privileges, or can change PHP code, or has access to the database
- ▶ Sensitive data isn't just health records, private credentials, or industrial secrets
- ▶ HUBzero stores `authorized_keys` files of privileged users elsewhere

STAY INFORMED

- ▶ Resources to follow and read in order to be aware of security threats
 - ▶ RedHat Network Alerts
 - ▶ Debian-Its mailing list
 - ▶ Debian-security-announce mailing list
 - ▶ CERTs in your area
 - ▶ CTSC-announce-inf-l mailing list
 - ▶ HUBzero lacks a way to inform you of important security patches (something we want to improve)



QUESTIONS?