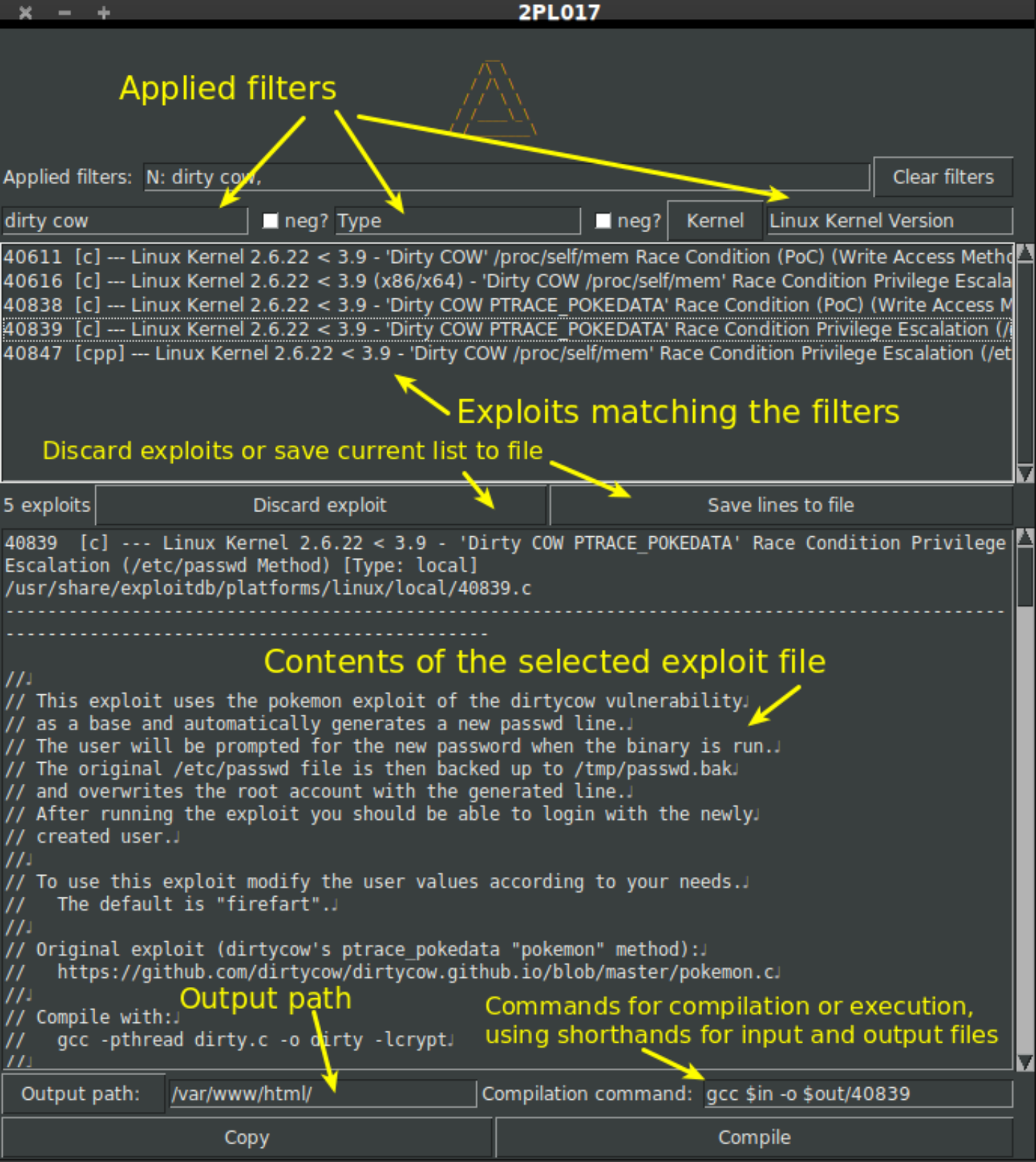
2PL017
Applied filters
Applied filters: N: dirty cow,
Clear filters
dirty cow
neg?
Type
neg?
Kernel
Linux Kernel Version
40611 [c] --- Linux Kernel 2.6.22 < 3.9 - 'Dirty COW' /proc/self/mem Race Condition (PoC) (Write Access Metho
40616 [c] --- Linux Kernel 2.6.22 < 3.9 (x86/x64) - 'Dirty COW /proc/self/mem' Race Condition Privilege Escala
40838 [c] --- Linux Kernel 2.6.22 < 3.9 - 'Dirty COW PTRACE_POKEDATA' Race Condition (PoC) (Write Access M
40839 [c] --- Linux Kernel 2.6.22 < 3.9 - 'Dirty COW PTRACE_POKEDATA' Race Condition Privilege Escalation (/
40847 [cpp] --- Linux Kernel 2.6.22 < 3.9 - 'Dirty COW /proc/self/mem' Race Condition Privilege Escalation (/et
Exploits matching the filters
Discard exploits or save current list to file
5 exploits
Discard exploit
Save lines to file
40839 [c] --- Linux Kernel 2.6.22 < 3.9 - 'Dirty COW PTRACE_POKEDATA' Race Condition Privilege Escalation (/etc/passwd Method) [Type: local]
/usr/share/exploitdb/platforms/linux/local/40839.c
Contents of the selected exploit file
// This exploit uses the pokemon exploit of the dirtycow vulnerability
// as a base and automatically generates a new passwd line.
// The user will be prompted for the new password when the binary is run.
// The original /etc/passwd file is then backed up to /tmp/passwd.bak
// and overwrites the root account with the generated line.
// After running the exploit you should be able to login with the newly
// created user.
// To use this exploit modify the user values according to your needs.
// The default is "firefart".
// Original exploit (dirtycow's ptrace_pokedata "pokemon" method):
// https://github.com/dirtycow/dirtycow.github.io/blob/master/pokemon.c
// Compile with:
// gcc -pthread dirty.c -o dirty -lcrypt
Output path
Commands for compilation or execution, using shorthands for input and output files
Output path:
/var/www/html/
Compilation command:
gcc $in -o $out/40839
Copy
Compile