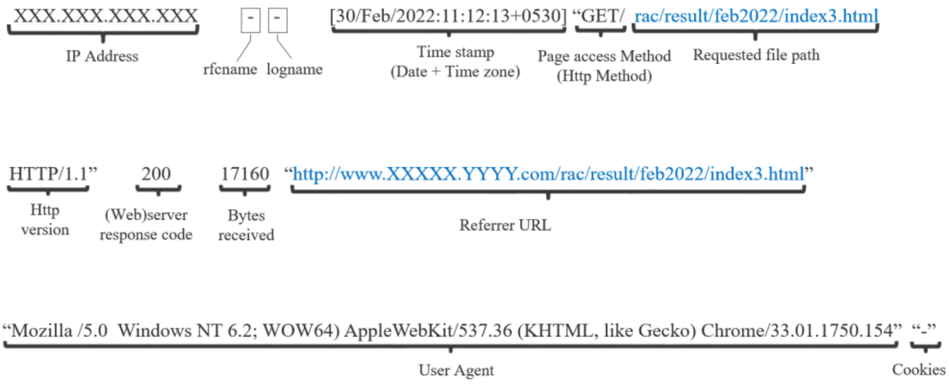
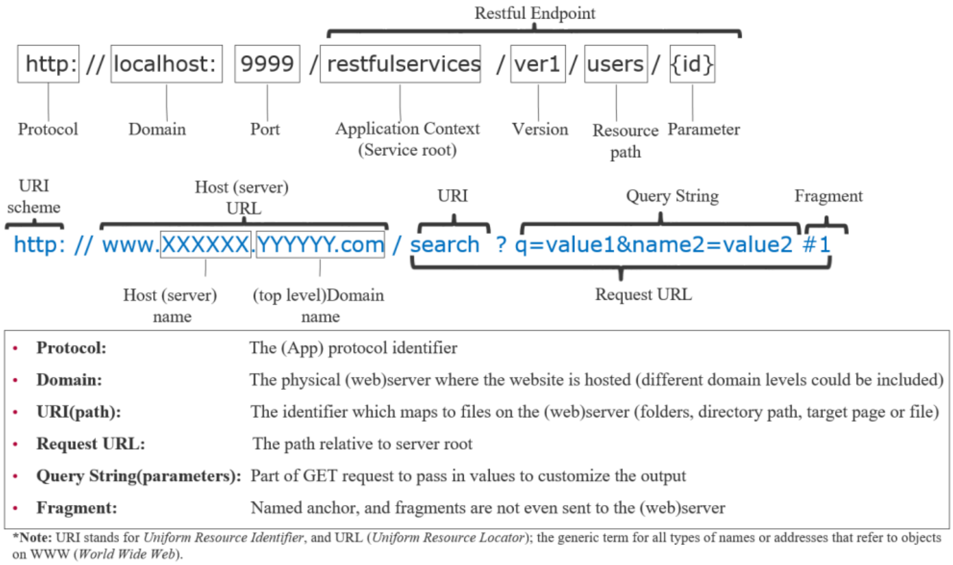


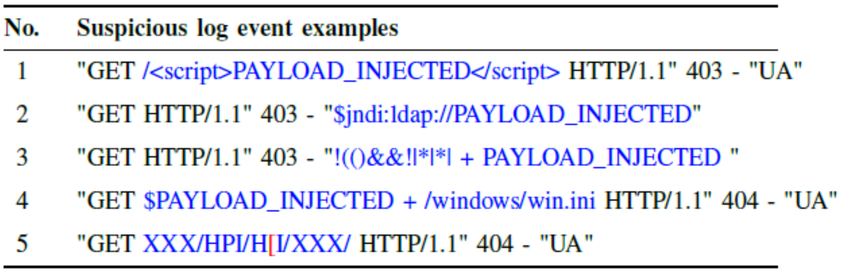
Anatomy of full Web server/proxy logs.



- **Protocol:** The (App) protocol identifier
- **Domain:** The physical (web)server where the website is hosted (different domain levels could be included)
- **URI(path):** The identifier which maps to files on the (web)server (folders, directory path, target page or file)
- **Request URL:** The path relative to server root
- **Query String(parameters):** Part of GET request to pass in values to customize the output
- **Fragment:** Named anchor, and fragments are not even sent to the (web)server

***Note:** URI stands for *Uniform Resource Identifier*, and URL (*Uniform Resource Locator*); the generic term for all types of names or addresses that refer to objects on WWW (*World Wide Web*).

General URL Anatomy



No.	Suspicious log event examples
1	"GET /<script>PAYLOAD_INJECTED</script> HTTP/1.1" 403 - "UA"
2	"GET HTTP/1.1" 403 - "$jndi:ldap://PAYLOAD_INJECTED"
3	"GET HTTP/1.1" 403 - "!(()&&!|*|*| + PAYLOAD_INJECTED "
4	"GET $PAYLOAD_INJECTED + /windows/win.ini HTTP/1.1" 404 - "UA"
5	"GET XXX/HPI/H[I/XXX/ HTTP/1.1" 404 - "UA"

Examples of web-server logs highlighting some attacks and exploitation of injections (e.g., XSS, SQLi, Log4j), where the attacker is attempting to manipulate the headers such as user agent (UA), referrer, and HTTP method. The last example is a representation of invalid characters (mostly unintentionally) in API within web-request possibly by the human factor to reach the end-point/service.

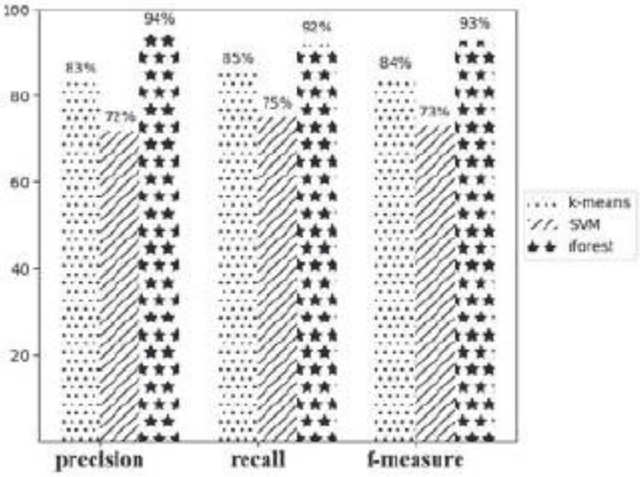


Fig. 5. Comparison of detection effects of different detection algorithms